

DIMM-MX257 Developer Kit for Windows Embedded CE 6

User Manual

Rev4 / 17.06.2011

All rights reserved. This documentation may not be photocopied or recorded on any electronic media without written approval. The information contained in this documentation is subject to change without prior notice. We assume no liability for erroneous information or its consequences. Trademarks used from other companies refer exclusively to the products of those companies. Microsoft, Windows, Windows95, Windows98, Windows NT, Windows 2000, Windows XP, Windows CE and MS-DOS are registered trademarks of the Microsoft Corporation.

Revision: **4 / 17.06.2011**

Rev	Date/Signature	Changes
1	22.06.10/bi	- First revision
2	09.07.10/bi	- Added Section about CAN driver
3	28.07.10/bi	- Added Camera Interface support
4	17.06.11/bi	- Renaming SYSDK to Developer Kit - Merging ADK and SYSDK Manual to one manual for the Developer Kit - Information about the BSP update added

This document is published by:

Emtrion GmbH
Greschbachstr. 12
D-76229 Karlsruhe
Tel: +49 (721) 62725-0
Fax: +49 (721) 62725-19
E-mail: mail@emtrion.de
Internet: <http://www.emtrion.de>

1 Index

Emtrion GmbH	3
2 Definitions.....	7
3 Terms and Conditions of License.....	7
4 Introduction.....	8
5 Requirements.....	9
5.1 Requirements on the host	9
5.2 SW-Requirements	9
5.3 HW-Requirements	9
5.4 Requirements on the target.....	9
6 Setting up the host	10
6.1 Installing the SDK for application development	10
6.2 Installing the Platform Builder for kernel development	10
6.3 Setting up the Ethernet (TCP/IP)	10
6.4 Setting up a TFTP server	11
6.5 Setting up a terminal.....	11
7 Setting up the target	12
7.1 Setting up for application development.....	12
7.2 Setting up kernel building or updating the image.....	12
8 DIMM-MX257 module variants.....	13
9 DVD of the Developer Kit.....	14
9.1 Installation of the BSP	14
9.2 Updating the BSP to the newest release	14
10 Features of the Developer Kit.....	15
10.1 Features presented by the BSP	15
10.1.1 Clocks.....	15
10.1.2 RAM	15
10.1.3 NAND	15
10.1.4 Display	16
10.1.5 I ² C	17
10.1.6 Touch	17
10.1.7 RTC.....	17
10.1.8 Ethernet.....	17
10.1.9 USB Host	17
10.1.10 USB Function.....	17
10.1.11 Serial Ports.....	18
10.1.12 Audio.....	19
10.1.13 CAN Bus Interface.....	19
10.1.14 SPI.....	19
10.1.15 Camera device	19
10.2 Features presented by the OSDesign.....	20
10.2.1 Persistent Registry.....	22

10.2.2	Autostart Mechanism	22
10.3	Software Development Kit (SDK)	23
10.4	Manuals	23
10.5	Features of the preconfigured OS Design	23
10.5.1	Applications – End User	23
10.5.2	Applications – End User	23
10.5.3	Communication Services and Networking.....	24
10.5.4	Core OS Services	24
10.5.5	Device Management	25
10.5.6	File System and Data Store	25
10.5.7	Fonts.....	26
10.5.8	Graphics and Multimedia Technologies	26
10.5.9	International.....	27
10.5.10	Internet Client Services.....	27
10.5.11	Security.....	27
10.5.12	Shell and User Interface.....	27
10.5.13	Device Driver	28
11	Generating a WinCE image (Kernel Development)	29
11.1	Settings of the Developer Kit kernel	29
11.1.1	Language settings.....	30
11.1.2	Build Options.....	30
11.1.3	Environment settings.....	30
11.1.4	Custom Build Actions.....	31
11.2	Build.....	32
11.2.1	The subproject ProjectRegistrySettings	32
11.2.2	Starting the build process	33
11.2.3	Incremental build	34
12	Licensing of the WINCE-Image	35
13	Downloading the image to the target.....	36
13.1	Perform download	36
14	Setting up the Platform Builder for target connection	37
15	Building a Debug WinCE-Image (Kernel Development).....	40
16	Data Exchange between Target and Workstation.....	41
16.1	Preparing the Desktop Computer	41
16.2	Preparing the Windows CE Device	41
16.3	USB ActiveSync	41
16.4	Remarks on Using ActiveSync	42
16.4.1	Data transfer.....	45
17	Application development	46
17.1	VS2005 for Application Development.....	46
17.1.1	Creating a new Managed Project	46
17.1.2	Deploying to the target.....	47
17.2	VS2008 for Application Development.....	49
18	Starting an application at system start	50
18.1	Emtrion’s autostart mechanism.....	50

18.2	Autostart with LaunchXX and DependXX.....	51
19	Sample Applications.....	53
19.1	SKitApp.....	53
19.1.1	How the sample application work	53
19.1.2	Changing the parameters via the menu	53
19.1.3	Changing the port.....	53
20	The Persistent Registry	55
20.1	Deleting the persistent registry	56
20.1.1	Deleting the registry by the bootloader	56
20.1.2	Deleting the registry by an application under Windows Embedded CE.....	56
21	Emtrion Tools for Windows Embedded CE	58
21.1	Writing the persistent registry	58
21.2	Getting the Version of the Operating System Kernel	58
21.3	SetIPAdr.....	58
22	Frequently Asked Questions (FAQ)	60
22.1	In General.....	60
23	Support.....	63
24	License Agreement	64
24.1	Emtrion License Agreement	64
24.2	Microsoft License Agreement	66
25	Literaturverzeichnis	70

2 Definitions

The table below lists some definitions of terms in this manual.

ActiveSync	An application from Microsoft used for the exchange of data between the Windows ® Embedded CE device and a desktop computer.
Applet	Application of the Control Panel which serves to control the system's properties.
Operating system kernel (OS kernel)	Complete Windows ® Embedded CE operation system running on the target platform. During the creation of the Windows ® Embedded CE OS kernel, modules and components may be omitted. This will create different OS kernels, which can each be found in an image file.
OSDesign	Project overview about all of the software components included in the kernel.
Desktop computer, Host	PC which fulfills the requirements according System Requirements (chapter 5.1)
Image, WINCE-Image (file)	File that is used by the bootloader and stored in the flash or RAM for execution
Persistent registry	Allows modifications to the registry which are still in effect after power off/on. A non-persistent registry of Windows ® Embedded CE is created from the image file every time the system is started up. Modifications made after system startup will not be in effect after a restart.
Software Development Kit (SDK)	Installable collection of header- and library files. The tools will be informed about the API functions supported by the associated OS kernel, and whether MFC, ActiveX etc. is supported.
Target platform	Target hardware where Windows ® Embedded CE is running.
DIMM-MX257	The target platform

3 Terms and Conditions of License

The DIMM-MX257Developer Kit for Windows ® Embedded CE comes with a runtime license for the Windows ® CE operating system and the associated drivers. Distribution of any product created by use of the Windows ® CE operating system and the associated drivers requires the purchase of separate licences.

The License Agreements to the software from Emtrion and Microsoft are stated in section License Agreements.

BY INSTALLING THE DEVELOPER KIT ON YOUR SYSTEM, YOU ARE AGREEING TO BE BOUND BY THE TERMS OF THOSE AGREEMENTS.

4 Introduction

The Developer Kit has been created to allow a simple and fast start-up of application development or building kernels with Windows® Embedded CE 6.0 R3 hardware platform.

Furthermore, it offers the possibility to reduce the time to market phase for software basing Windows® Embedded CE 6.0 R3 on DIMM-MX257.

For developing applications, the Developer Kit contains a preconfigured operating system kernel. This kernel provides a wide range of components and functions of Microsoft's popular Windows® Embedded CE Win32 APIs and has a size of about 25MB, which allows it to be used for a wide range of applications.

The preconfigured kernels allows to write applications without creating own operating system kernels.

A detailed description of the operating system kernels can be found in the section **Fehler! Verweisquelle konnte nicht gefunden werden..**

For building kernels, the Developer Kit contains all the necessary software components that are needed for build kernels on DIMM-MX257.

Specifically tailored operating system kernels may include, for example, special fonts, keyboard layout or additional drivers. These components will either be supplied by the customer or designed by emtrion, if desired.

5 Requirements

5.1 Requirements on the host

The requirements are dependent on whether you want to develop applications only or build also kernels. The minimal requirements on the host are listed in the following subchapters.

5.2 SW-Requirements

- Operating system
 - o Windows® XP SP3 or later
- Microsoft ActiveSync Version 4.2 for development purpose and data exchange between target and host
- Microsoft Visual Studio 2005. A 180-Day Trial Version is delivery by the Developer Kit.
- When you want to update the image a TFTP-Server is required too. The TFTP-Server is used for downloading of the WINCE-Image to the target. A TFTP-Server is not part of the Developer Kit, but can be downloaded for free for example at http://tftpd32.jounin.net/tftpd32_download.html. When you are using this tftp, be sure it's DHCP service is disabled to avoid conflicts with the current DHCP server at you network.

For building kernels you additional require

- Windows Embedded CE R3 Platform Builder. For them a 180-Day Trial Version is also delivered with the Developer Kit.

5.3 HW-Requirements

- CPU-Speed: 933 MHz, recommended 2 GHz
- RAM: 512 MB, recommended 2 GB
- Interfaces
 - o 1x Ethernet port
 - o 1x RS232 (serial port)
 - o 1x USB 1.1 or 2.0
- DVD-ROM drive
- If you want to create your own OS images, up to 18GB free hard disk space for
 - o Installation of BSP of the Developer Kit
 - o Platform Builder Development environment (Microsoft Windows® Embedded CE 6 R3) if not installed already

5.4 Requirements on the target

For the target the following devices are recommended:

- USB keyboard
- USB mouse

6 Setting up the host

At this point we assume that the trial or full version of Microsoft® Visual Studio® 2005 with SP1 and Microsoft® ActiveSync are already installed on the host.

6.1 Installing the SDK for application development

The SDK allows you to develop applications based on the delivered kernel. To make this possible install the SDK when VS2005 is not open. You can install it from the DVD. During the installation, the SDK is added to the list "Installed SDKs" from VS2005. At this point HICODIMM7723 can be chosen in VS2005 for generating applications on the DIMM-MX257.

Note:
HiCO.DIMM7723 is the old name of DIMM-MX257.

6.2 Installing the Platform Builder for kernel development

Note:
This step is only necessary when you want building kernels.

When you install Microsoft® Windows Embedded CE 6 it is recommended to install only the features for the used CPU. For this Developer Kit you need the CPU type ARM.

First install the trial or full version of Microsoft® Windows Embedded CE 6 on the host. Follow the instructions displayed by the installer of Microsoft Windows Embedded CE 6.0. Platform Builder 6.0, which is a development environment, is also installed when Microsoft Windows Embedded CE 6.0 is installed. The Platform Builder 6.0 is a Plug-In for Microsoft Visual Studio 2005.

To get a Microsoft® Windows Embedded CE 6.0 R3 environment, install the following in the given order:

1. Windows Embedded CE 6.0 Platform Builder Service Pack 1 (required if PB 6.0 Tools have been installed).
2. The trial or full version of Windows Embedded CE 6.0 R2
3. Windows Embedded CE Cumulative Product Update Rollup 12/31/2008
4. All updates for Windows Embedded CE 6.0 R2 release in 2009
5. Install the Windows Embedded CE 6.0 R3 update.
6. Install all updates for Windows Embedded CE 6.0 R3.

Note:
Do NOT install older update packages after Windows Embedded CE 6.0 R3 has been installed.

6.3 Setting up the Ethernet (TCP/IP)

TCP/IP is used for Ethernet connection to DIMM-MX257. Make sure that TCP/IP is already setup for the network communication at the host.

6.4 Setting up a TFTP server

A TFTP server is required to download or update the image on the target. Make sure the TFTP is running and its search directory points to the location of the image. A link for a TFTP server is stated in the section "SW-Requirements".

6.5 Setting up a terminal

As terminal software, use the terminal software that comes with your host or any software you are already using. In the terminal setting window, set the communication parameters as follow:

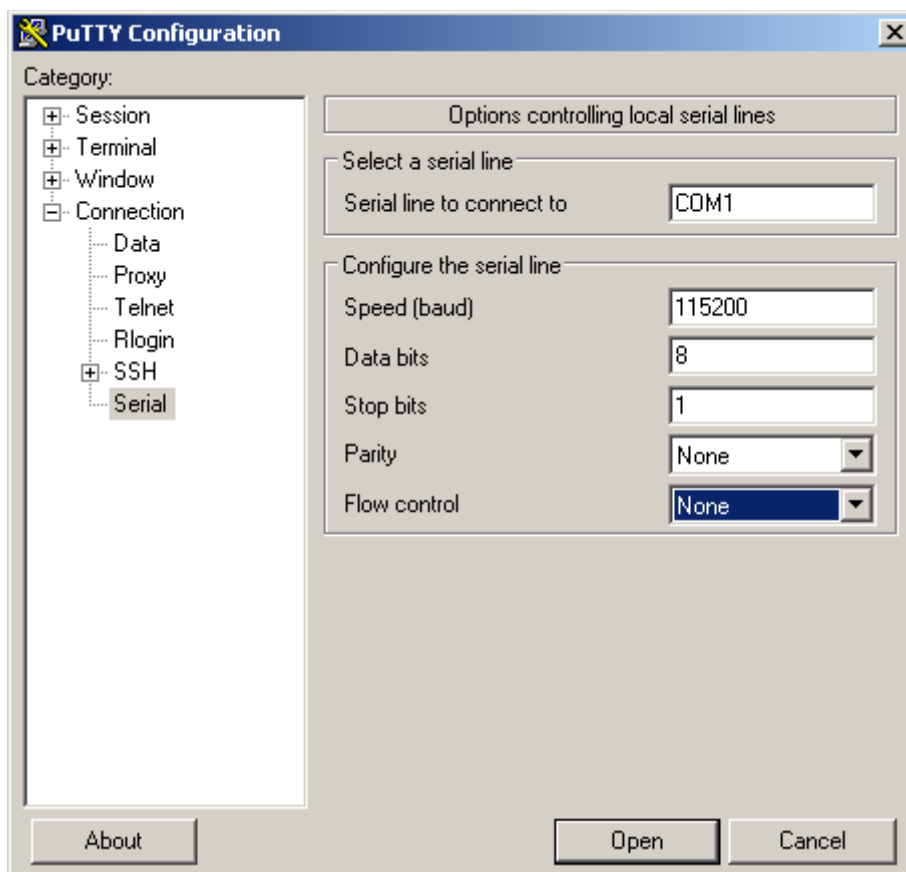
Baud rate 115200 bps

Data length 8 bits

Parity none

Stop bit 1

Flow control none



Use the supplied serial null-modem cable to connect serial port COM1 (UART A) on the target to the host.

7 Setting up the target

This Developer Kit is delivered with the core module DIMM-MX53 and dependent on the scope of the delivery with one of the available emtrion DIMM-base boards. To make a fast operation possible the target is already assembled. For developing purpose only a few steps are necessary.

Please note that everything is discharged when touching the target.

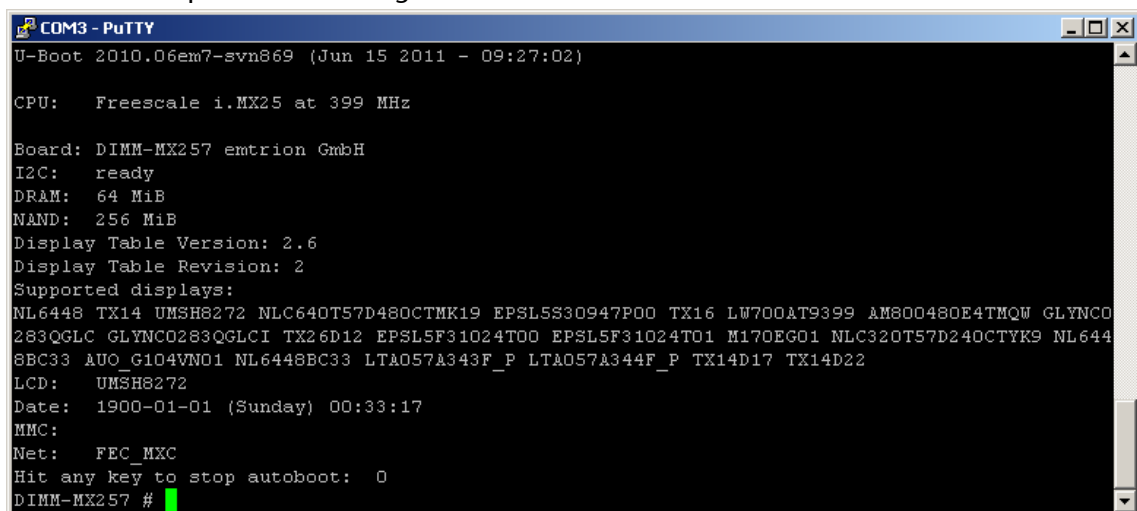
7.1 Setting up for application development

There are only two steps to do:

- Connect the target to the host via the supplied USB cable A-B to allow a Microsoft® ActiveSync (Mobile Device Center) connection. The USB-B type connector is located at the front of the target.
- Use the Ethernet cable to connect the target to the network where the host is linked to.

7.2 Setting up kernel building or updating the image

- A Microsoft ActiveSync connection is not necessary
- Enter the bootloader prompt as described in the Bootloader Manual.
- Connect the target to the host via the serial cable. Use the 9-pin D-Sub jack of the serial port UART A at the target.
- Power ON or reset the target. When a terminal application is running on the host the bootloader outputs the following information:



```
COM3 - PuTTY
U-Boot 2010.06em7-svn869 (Jun 15 2011 - 09:27:02)

CPU:   Freescale i.MX25 at 399 MHz

Board: DIMM-MX257 emtrion GmbH
I2C:   ready
DRAM:  64 MiB
NAND:  256 MiB
Display Table Version: 2.6
Display Table Revision: 2
Supported displays:
NL6448 TX14 UMSH8272 NLC640T57D480CTMK19 EPSL5S30947P00 TX16 LW700AT9399 AM800480E4TMQW GLYNCO
283QGLC GLYNCO283QGLCI TX26D12 EPSL5F31024T00 EPSL5F31024T01 M170EG01 NLC320T57D240CTYK9 NL644
8BC33 AUO_G104VNO1 NL6448BC33 LTA057A343F_P LTA057A344F_P TX14D17 TX14D22
LCD:   UMSH8272
Date:  1900-01-01 (Sunday) 00:33:17
MMC:
Net:   FEC_MXC
Hit any key to stop autoboot:  0
DIMM-MX257 #
```

- Make a physical Ethernet connection, like in application development above. Set the network environment variables to your requirements. How to update the installed images is described in the Bootloader Manual.

8 DIMM-MX257 module variants

Please regard, that there are two variants of the DIMM-MX257, the standard variant and the camera interface-enabled variant. They differ in the available interfaces.

The standard variant has 4 UART interfaces and 2 SD Card interfaces, whereas the camera variant has only 3 UART interfaces, 1 SD Card interface, but an additional camera interface available. As a result, there are also two variants of the Windows CE kernel images. If you update the image, please make sure that you use the right one to be able to use all interfaces of your module.

The two image variants are named as follows on the ADK DVD:

Standard variant	wince-dimm_mx257
Camera variant	wince-dimm_mx257_cam

For details on how to download a kernel image onto the device please see [Downloading the image to the target](#).

9 DVD of the Developer Kit

The supplied DVD contains all the software and documentation you need to start developing applications or building kernels based on Windows® Embedded CE 6.0 R3 for DIMM-MX257.

The DVD contains all data required for development of applications and/or own images for the platform DIMM-MX257. In the root directory of the DVD you will find a html-file named start_here.html. This file can be opened in a html browser such as Internet Explorer or Firefox. The page which is shown gives an overview about the DVD contents.

9.1 Installation of the BSP

Before installing the BSP of the Developer Kit, please make sure that Microsoft® Windows Embedded CE 6.0 R3 is already installed on the host (see chapter “Installing the Platform Builder for kernel development”).

The installation of the BSP is a copy operation. Use the Windows Explorer (or another file explorer) to copy the contents of the directory BSP from the DVD to the location of your Windows Embedded CE installation. The folder WINCE600 is the same folder as the folder WINCE600 which was created during the installation of the Platform Builder. You should preserve the directory structure of the copied data.

Each folder in the copied structure contains a subfolder with the name .svn. These folders are not used by the Platform Builder but are required if you want to download updates from the emtrion update server (see next chapter).

9.2 Updating the BSP to the newest release

Emtrion provides the possibility to update your BSP to the newest revision which was released from emtrion. This can be done using a subversion client. Subversion is a open-source version control system. More details you can find in the document “How to update your product from the update repository”. This document is also on the DVD.

10 Features of the Developer Kit

The Developer Kit is delivered with the DIMM-MX257 and a base board. Together they provide a wide range of functionality. The following sections describe what the software is supporting at release time of this manual.

10.1 Features presented by the BSP

The BSP presents the foundation for building OS Designs on DIMM-MX257. It provides the support of features on DIMM-MX257. The BSP is delivered with binary files. The supported features are listed below.

10.1.1 Clocks

The input clock of the processor is 24,576MHz . This results in the following clocks.

CPU	→ 399MHz
DDR SDRAMs	→ 133MHz
Bus	→ 66,5MHz

10.1.2 RAM

DIMM-MX257 is equipped with **64MB DDR SDRAM**. The data width is 32Bit. The following figure shows the memory layout of the SDRAM under WinCE.

<u>Physical address</u>	<u>Usage of the RAM areas</u>
0x80000000	Reserved
0x80001000	Structure _DRIVER_GLOBALS used as interface Bootloader/Wince
0x80002000	Reserved
0x80100000	WINCE-Image(kernel)
0x81100000	Object-Store, RAM for WINCE
0x83C00000	Bootloader/Frame Buffer
0x83FFFFFF	

10.1.3 NAND

The NAND MT29F2G from Micron serves as storage on the DIMM-MX257 module. It has a size of 256 MB and contains the U-Boot bootloader together with its configuration as well as a WinCE Image, which can be booted by the bootloader. It also contains a FlashFile-System, which is used under WinCE.

NAND address Usage of the NAND

0x00000000	Bootstrapper
0x00020000	U-Boot
0x000A0000	NVRAM Environment variables of the bootloader
0x000E0000	Splash-Screen
0x00120000	Reserved
0x00200000	WINCE-Image
0x04000000	FlashFile-System (exFAT)
0x0FFFFFFF	

At power on or following a reset the WINCE-Image is copied from NAND to RAM by the bootloader to be started.

The storage of the FlashFile-System appears as folder "NAND Flash" inside the WINCE.

10.1.4 Display

The display driver is prepared for the TFT displays in the table below.

Display	Resolution	Colours	Remarks
TFT, Admatec NLC640T57D480CTMK	640x480	16bpp	Name for display NLC640T57D480CTMK19
TFT, Hitachi TX14D11VM1CBA	320x240	16bpp	Name for display TX14
TFT, UMSH8272	800x480	16bpp	Name for display UMSH8272
TFT, Hitachi TX26D12VM0APA	800x600	16bpp	Name for display TX26

To offer a wide flexibility in changing displays the used display has to be selected by the bootloader. For them look to the section [Changing the display](#).

10.1.5 I²C

The DIMM-MX257 possesses a I²C-Interface connected to a bus. Several slave devices can be connected to the I2C-Bus, like audio codec and external RTC, depending on the base board. They are serviced by their own device driver.

But it is possible to connect additional devices to the bus. In this case an API is available. Since the driver is provided from Freescale please refer to their documentation. (1)

10.1.6 Touch

The DIMM-MX257 has an integrated 4-wire touch controller. Depending on the display, which is configured for the U-Boot, U-boot passes an initial touch screen calibration to Windows CE at boot.

Recalibration under Windows CE can be done via the calibration applet "Stylus" in the control panel.

10.1.7 RTC

Due to the RTC of the DIMM-MX257 is not battery buffered, there is an external RTC on the DIMM-MX257 which is buffered by the battery on any suited base board. The external RTC is connected to the I2C-Bus. Its slave address is 0x68.

When starting WINCE, the contents of the external RTC are read and then written to the internal RTC. While WINCE is running the internal RTC is used. But if a new time/date is set, the external RTC is also updated.

Using the WIN32-API functions GetSystemTime and SetSystemTime you can read and set the time of the RTC from your application.

10.1.8 Ethernet

With the implemented ethernet driver the provided network functionality of WINCE is available. The Ethernet driver supports in detail

- 100/10MBit full- and half-duplex
- Full-duplex flow control
- Auto-negotiation

10.1.9 USB Host

The USB Host controller was successfully tested with following devices under Windows CE.

- keyboard
- mouse
- memory stick
- hub

Hubs can also be cascaded.

10.1.10 USB Function

The i.MX257 has an integrated USB OTG controller. On the DIMM-MX257 it is setup to be used as USB Function-only. It delivers the functionality for application developers to use Microsoft ActiveSync for debugging purposes.

Microsoft ActiveSync offers to you

- Debugging in combination with Visual Studio 2005/2008
- Using the Remote Tools of VS2005/VS2008
- Exploring the target

10.1.11 Serial Ports

The DIMM-MX257 has four serial communication interfaces. One features RTS/CTS signals and RS232 signal level. The others only provide the RX/TX signals and LVTTTL level.

The relationship between the names and serial ports from the hardware and software point of view and some more supported features are summarised in the table below.

Software	COM2	COM5	COM1	COM3
Interface	UART2	UART5	UART1	UART3
Signals	LVTTTL*	LVTTTL*	RS232	LVTTTL*
HiCO.DIMM-Base UART-x Connector	UART-E pin header J15 **	UART-D pin header J15 **	UART-A D-Sub (bottom)	UART-B D-Sub (top)
HiCO.DIMM-ECObase UART-x Connector on	UART-E pin header J12 **	UART-D pin header J12 **	UART-A D-Sub	UART-C pin header J12 **
HW-Handshake RTS/CTS	No	No	Yes	No
Hardware Flow Control RTS/CTS	No	No	No	No
HW-FiFo				
- Receive	32Byte	32Byte	32Byte	32Byte
- Transmit	32Byte	32Byte	32Byte	32Byte
Baudrates				
- 1200	yes	yes	yes	yes
- 2400	yes	yes	yes	yes
- 4800	yes	yes	yes	yes
- 9600	yes	yes	yes	yes
- 14400	yes	yes	yes	yes
- 19200	yes	yes	yes	yes
- 38400	yes	yes	yes	yes
- 57600	yes	yes	yes	yes
- 115200	yes	yes	yes	yes

- > 115200	***	***	***	***
------------	-----	-----	-----	-----

(*) Suitable RS232-adapters are available from emtrion

(**) For more details please refer to the corresponding HW- manual

(***) Only on request

All the serial ports can be accessed by the WIN32-API of WINCE. But with the debug version of the Developer Kit kernel, **COM1** is reserved for debug purpose.

10.1.12 Audio

The SSI-interface of the i.MX257 is used to make audio available with any of the suited base boards from emtrion. The base boards are equipped with the audio codec TLV320AIC23B. The codec is connected to the I2C-Bus. Its slave address is **0x1B**.

The driver can process 8- and 16bit samples in mono and stereo for sample rates up to 44100Hz.

10.1.13 CAN Bus Interface

The i.MX257 features two internal CAN-Controllers. Freescale offers an own driver for these controllers, but for compatibility reasons regarding our existing CAN Architecture and for performance reasons emtrion developed an own driver.

This driver supports the HiCOCAN API, and its usage is documented in detail in this document (2).

10.1.14 SPI

The DIMM-MX257 features 1 SPI interface on the DIMM connector. The driver for it is provided by Freescale. To learn about the API please refer to Freescale's documentation (1).

10.1.15 Camera device

The camera-variant of the DIMM-MX257 features the digital camera interface on the DIMM connector. The camera device driver is based on the driver provided by Freescale. To learn about the API please refer to Freescale's documentation (1).

It was extended to support two CMOS camera sensors: Omnivision OV7720 and Omnivision OV2640.

The attached sensor can be specified by following registry key:

HKEY_LOCAL_MACHINE\Drivers\BuiltIn\Camera\Camerad (REG_DWORD)

Value	Selected Camera
0	Omnivision OV7720
1	Omnivision OV2640

If you need a driver for another camera sensor, feel free to contact emtrion.

10.2 Features presented by the OSDesign

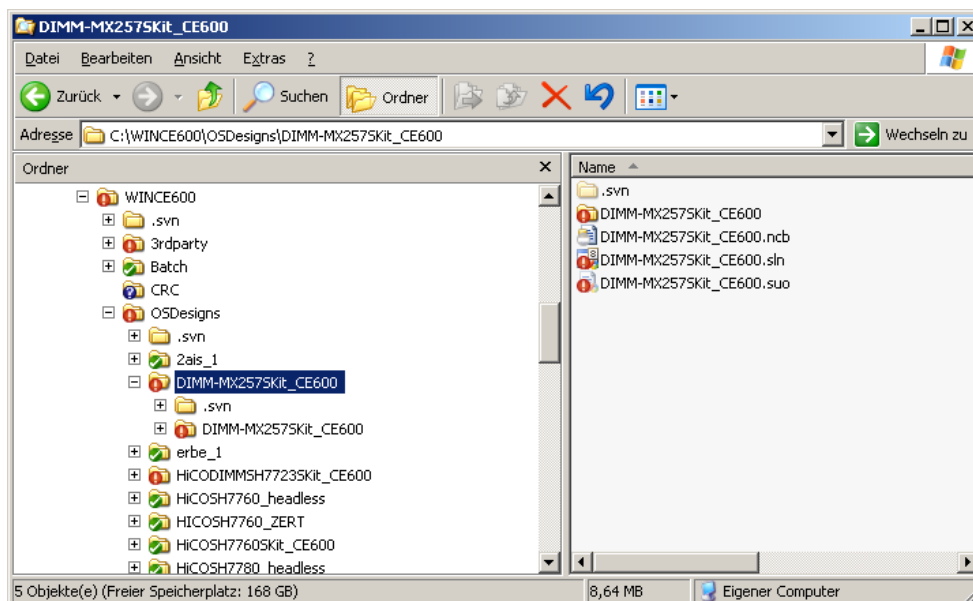
The OSDesign is the summary of all the software components that have been included into the kernel of the Developer Kit and reflects its functionality as well. The OSDesign comprises the BSP, some emtrion tools and selective functions delivered by Windows Embedded CE 6.0 R3. Here are some highlights of the kernel.

Highlights of the Developer Kit kernel

- .NET Compact Framework 3.5
- CAB File Installer/Uninstaller
- Microsoft Flash File System
- Autostart of a custom application from Mass-Storage-Devices (e.g. Flash File System, USB Stick, ...)
- Graphical User Interface
- Complete Win32-API, available for Windows CE
- Wired Local Area Network
- USB Host (USB keyboard, USB mouse, USB memory stick and USB printer)
- Touch
- 4 serial ports (COM1:, COM2:, COM3:, COM4: and COM5:)
- USB function (ActiveSync support)
- Internet Explorer 6.0 for Windows Embedded CE
- Telnet and FTP server functionality
- Software development and debugging with Visual Studio 2005/2008 via Ethernet is prepared (ConmanClient2.exe and CMAccept.exe are integrated into the kernel)

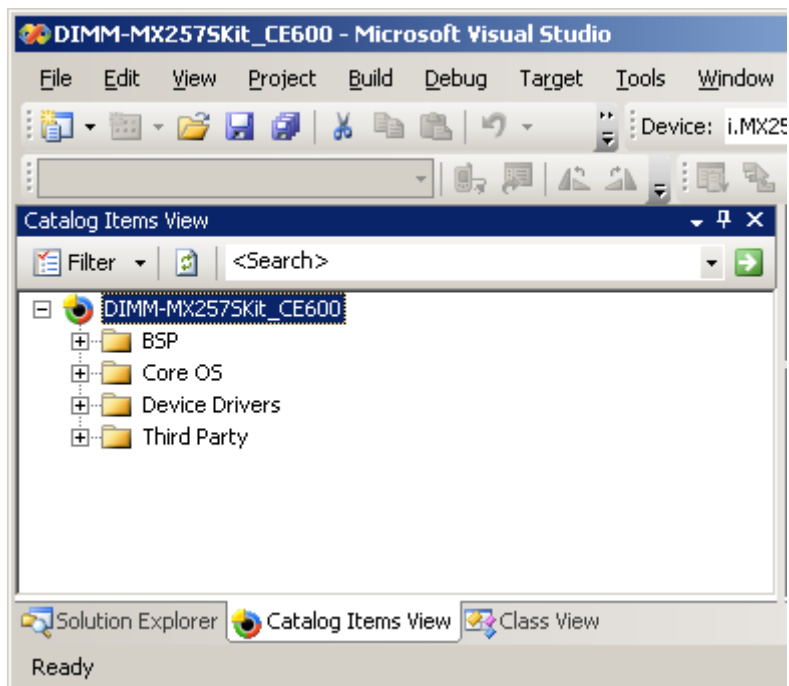
The exact functionality what is containing the kernel can be checked in the window "Catalog Items View" of the OSDesign DIMM-MX257Skit_CE600.

The OSDesign can be opened either out of the VS2005 by navigating to the location of the solution or by double click the solution file **DIMM-MX257SKit_CE600.sln**.

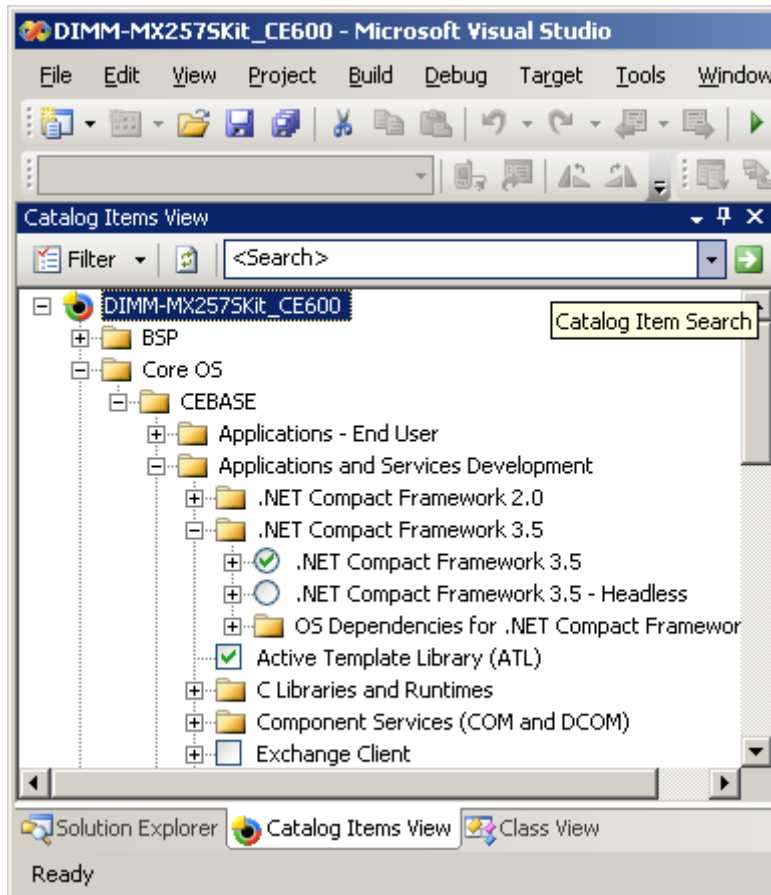


The figure below presents the opened solution of the OSDesign (**DIMM-MX257SKit_CE600**) of the Developer Kit and the "Catalog Items View" window in VS2005. There are four folders visible. The OSDesign of the Developer Kit consists of components from the folders Core OS, Device

Drivers and Third Party. Look in any of the folders for selected items to get an overview of the components in the kernel.



The next screen shows a part of the opened folder CEBASE and some of its items. The selected components are part of the kernel.



10.2.1 Persistent Registry

The OSDesign of the Developer Kit kernel comes with a hive-based persistent registry. At such a kernel, changes on the registry can be made persistent. This means the changes are kept at power off or at reset.

But be careful with wrong settings on the registry. This can cause the kernel to not start correctly the next time. When this behaviour occurs deleting the persistent registry is the only way to leave this issue. This can be done by setting a flag in the bootloader's environment variables. To avoid such situations, kernels without a persistent registry can also be ordered at emtrion.

For more details to the persistent registry see at the section [Persistent Registry](#).

10.2.2 Autostart Mechanism

In the OSDesign an autostart feature is implemented. It allows the execution of a specified application at system start. There are two options, starting the application at launch time of the explorer or at launch time of the autostart program.

More information at this topic is given by the chapter [Starting an application at system start](#).

10.3 Software Development Kit (SDK)

Together with the Developer Kit kernel, the suitable Software Development Kit for VS2005 is delivered too. This allows you to develop applications based on the Developer Kit kernel.

Note:

Normally, a SDK is valid for a specific OSDesign. But it is possible to develop applications based on a SDK different to an OSDesign. In such a case you must be sure the application uses only APIs that are supported by the different OSDesign.

To avoid errors caused by used functions that are not supported by the OSDesign, we recommend to use the appropriate SDK at any time.

You can install the SDK from the Developer Kit DVD.

10.4 Manuals

There are some manuals delivered by this Developer Kit. They are stored in the folder "documents" of the DVD.

10.5 Features of the preconfigured OS Design

This chapter describes briefly the contents of the kernel delivered on the DVD or available online in the product update repository or on our support site.

Note:

The image on our support site is not licensed in contrast to the image which can be found on the DVD or in the product update repository.

The kernel is a typical one that should be suitable for most of applications. If modifications are necessary, you may consider to make your own image or just asking emtrion to design an individual kernel for your needs.

10.5.1 Applications – End User

	Included
ActiveSync	X
File Sync	X
CAB File Installer/Uninstaller	X

10.5.2 Applications – End User

	Included
.NET Compact Framework 2.0	
.NET CompactFramework 2.0	-
.NET CompactFramework 2.0 - headless	-
OS Dependencies for .NET CF 2.0	-
OS Dependencies for .NET CF 2.0 - headless	-
.NET Compact Framework 3.5	
.NET CompactFramework 3.5	X
.NET CompactFramework 3.5 - headless	-
OS Dependencies for .NET CF 3.5	X
OS Dependencies for .NET CF 3.5 - headless	-

Active Template Library (ATL)	X
C Libraries and Runtimes	
C++ Runtime Support for Exception Handling and Runtime Type Information	X
Full C Runtime	X
Standard I/O (STDIO)	X
Standard I/O ASCII (STDIOA)	X
Standard String Functions - ASCII	X
COM and DCOM	
DCOM, incl. COM Storage	X
Minimal COM (No OLE Support)	-
String Safe Utility Functions	X
XML	
XML Core Services and Document Object Model	X
XML Query Languages (XQL)	X

10.5.3 Communication Services and Networking

	Included
Networking General	
NDIS User-mode I/O Protocol Driver	X
Network Driver Architecture (NDIS)	X
Network Utilities (Ipconfig, ping, route, netstat)	X
TCP/IP	X
IP Helper API	X
TCP/IPv6 Support	X
Windows Networking API/Redirector (SMB/CIFS)	-
Winsock Support	X
Networking – Local Area Network (LAN)	
Wired Local Area Network (802.3, 802.5)	X
Networking – Wide Area Network (WAN)	
Dial Up Networking (RAS/PPP)	X
AutoDial	X
Standard Modem Support for Dial Up Networking	X
Telephony API (TAPI 2.0)	X
Unimodem Support	X
Servers	
Core Server Support	X
File Server (SMB/CIFS)	-
FTP Server	X
RAS Server/PPTP Server (Incoming)	-
SNTP Client with DST	X
Telnet Server	X

10.5.4 Core OS Services

	Included
Toolhelp API	X
Device Manager	X

Display Support	X
Serial Port Support	X
UI Proxy for Kernel-Mode Drivers	X
Windows Embedded CE Driver Development Kit Support Library	X
Kernel Functionality	
Fiber API	X
FormatMessage API	X
FormatMessage API - System Error Messages	X
Memory Mapped Files	X
Message Queue - Point-to-Point	X
Target Control Support (Shell.exe)	X
Notification	
Non UI based Notification	-
UI based Notification	X
Power Management	
Power Management (Full)	X
Power Management (Minimal)	-
USB Support	
USB Function Driver	X
USB Host Support	X
USB Human Input Device (HID) Class Driver	X
USB HID Keyboard and Mouse	X
USB HID Keyboard only	X
USB HID Mouse only	X
USB Printer Class Driver (PCL3)	X
USB Storage Class Driver	X

10.5.5 Device Management

	Included
Simple Network Management Protocol (SNMP)	X
Device Management Client	X

10.5.6 File System and Data Store

	Included
Compression	X
Database Support	
CEDB Database Engine	X
File and Database Replication, bit-based	X
File Cache Manager	-
File System - Internal	
RAM and ROM File System	X
ROM-only File System	-
Registry Storage	
Hive-based Registry	-
RAM-based Registry	X
Storage Manager	
FAT File System	X
Partition Driver	X

Release Directory File System	X
Storage Manager Control Applet	X
System Password	X

10.5.7 Fonts

	Included
Arial (Subset 1_30)	X
Courier New (Subset 1_30)	X
Symbol	X
Tahoma (Subset 1_30)	X
Tahoma Bold	X
Wingding	X

10.5.8 Graphics and Multimedia Technologies

	Included
Audio	
Audio Compression Manager	X
Waveform Audio	X
Graphics	
Direct3D Mobile	X
DirectDraw	X
Gradient Fill Support	X
Still Image Codec Support (Encode and Decode)	X
Still Image Encoders and Decoders	X
BMP Decoder	X
GIF Decoder	X
JPG Decoder	X
PNG Decoder	X
Media	
Audio Codecs and Renderers	
MP3 Codec	X
MPEG-1 Layer 1 and 2 Audio Codec	X
MS ADPCM Audio Codec	X
Wave/AIFF/au/snd File Parser	X
Waveform Audio Renderer	X
WMA Codec	X
DirectShow	
ACM Wrapper Filter	X
DirectShow Core	X
DirectShow Display	X
DirectShow Error Messages	X
DirectShow Video Capture	X
DMO Wrapper Filter	X
Media Formats	
MPEG-1 Parser/Splitter	X
Video Codecs and Renderers	
Direct Show Video Renderer	X

MPEG-1 Video Codec	X
MS RLE Video Codec	X
Video/Image Compression Manager	X
WMV/MPEG-4 Video Codec	X
Windows Media Player	
Windows Media Player	X
Windows Media Player OCX	X
Windows Media Technologies	X

10.5.9 International

	Included
Input Method Manager (IMM)	X
Local Services	
English (US) National Language Support only	-
National Language Support (NLS)	X

10.5.10 Internet Client Services

	Included
Internet Explorer 6.0 for Windows Embedded CE - Standard Components	X
Internet Explorer 6.0 Sample Browser	X
Internet Explorer 6.0 for Windows Embedded CE Components	X
Internet Explorer Browser Control Host	X
Internet Explorer HTML/DHTML API	X
Internet Explorer Multiple-Language Base API	X
URL Moniker Services	X
Windows Internet Services	X
Internet Options Control Panel	X
JScript 5.6	X
VBScript 5.6	X

10.5.11 Security

	Included
Authentication Services	X
NTLM	-
Schannel (SSL/TLS)	X
Credential Manager	X
Cryptography Services (CryptoAPI 1.0) with High Encryption Provider	X
Certificates (CryptoAPI 2.0)	X

10.5.12 Shell and User Interface

	Included
Graphics, Windowing and Events (GWES)	

Minimal GDI Configuration	X
Minimal GWES Configuration	X
Minimal Input Configuration	X
Minimal Window Manager Configuration	X
Command Shell	
Aygshell API Set	X
Command Processor	X
Console Window	X
Graphical Shell	
Standard Shell	X
Common Controls	
Common Control	X
Common Dialog Support	X
Control Panel Applets	X
Mouse	X
Network User Interface	X
Software-based Input Panel (SIP)	X
SIP for Small Screens	X
Software-based Input Panel Driver	X
Touch Screen (Stylus)	X

10.5.13 Device Driver

	Included
Analog Digital Convertor	X
TLV320AIC23 Stereo Audio	X
Backlight LCDC	X
Fake Battery	X
CSPI Bus	X
U-Boot Display Configuration	X
Emtrion Driver Keyboard	X
Fast Ethernet Controller	X
GPT (General Purpose Timer) 1	X
I2C Bus	X
PWM	X
Enhanced SD Host Controller	X
UART 1-3,5	X
Touchscreen	X
USB High Speed OTG Pure Client Function	X
USB High Speed Host Port Function	X
Hive based Registry stored in NAND Flash	X
Micron MT29F2G08 NAND Flash	X
HiCOCAN High-level Driver	X
HiCOCAN Platform Driver for DIMM-MX257	X

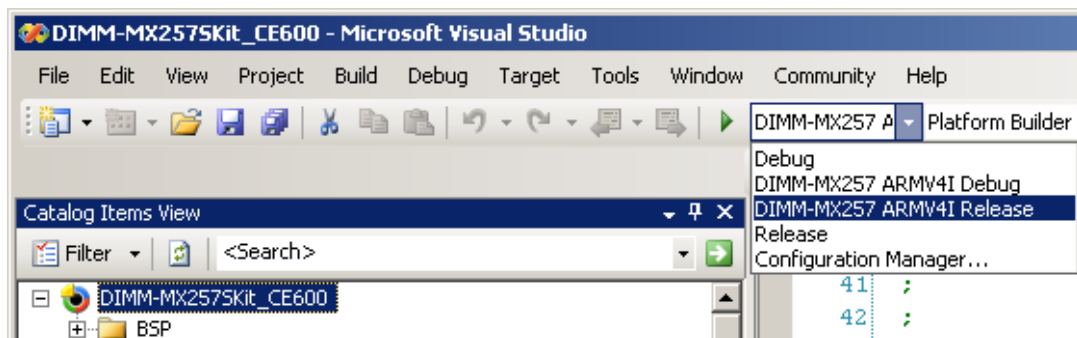
11 Generating a WinCE image (Kernel Development)

Normally, when you generate the first image after creating or opening a new OSDesign, make build-related settings. The settings for the OSDesign of the Developer Kit have already set and differ from release and debug version.

This section describes how to generate an image for release of the Developer Kit kernel. If you wish to generate an image for debug, refer to section Debug build.

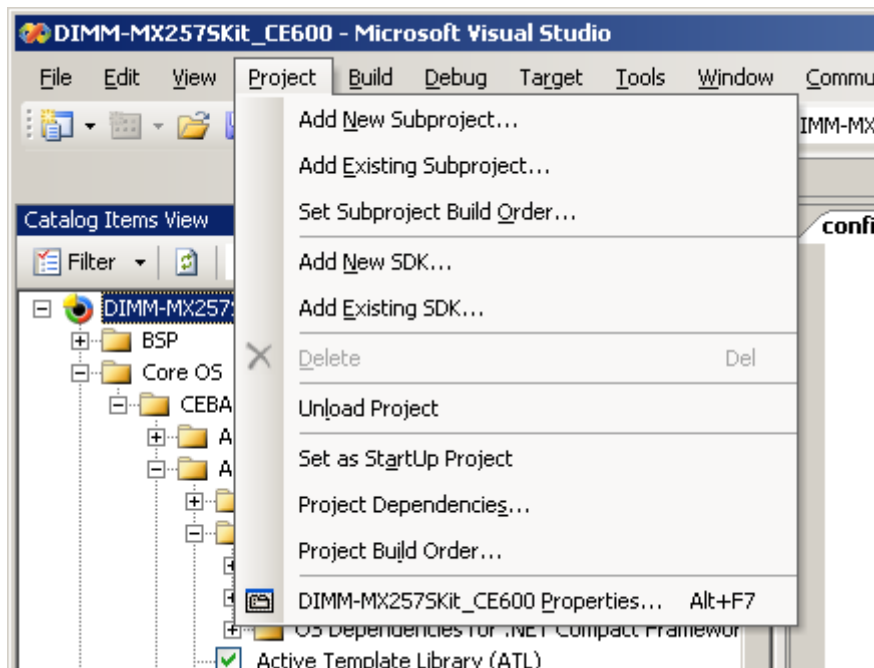
At this point we assume the solution of the OSDesign is opened.

- ❖ First, choose the configuration for release as shown below.



- ❖ Second, navigate to the Project Properties Pages

Go to "Project" menu of VS2005 and select "DIMM-MX257Skit_CE600 Properties ..." to open the Project Properties Pages.

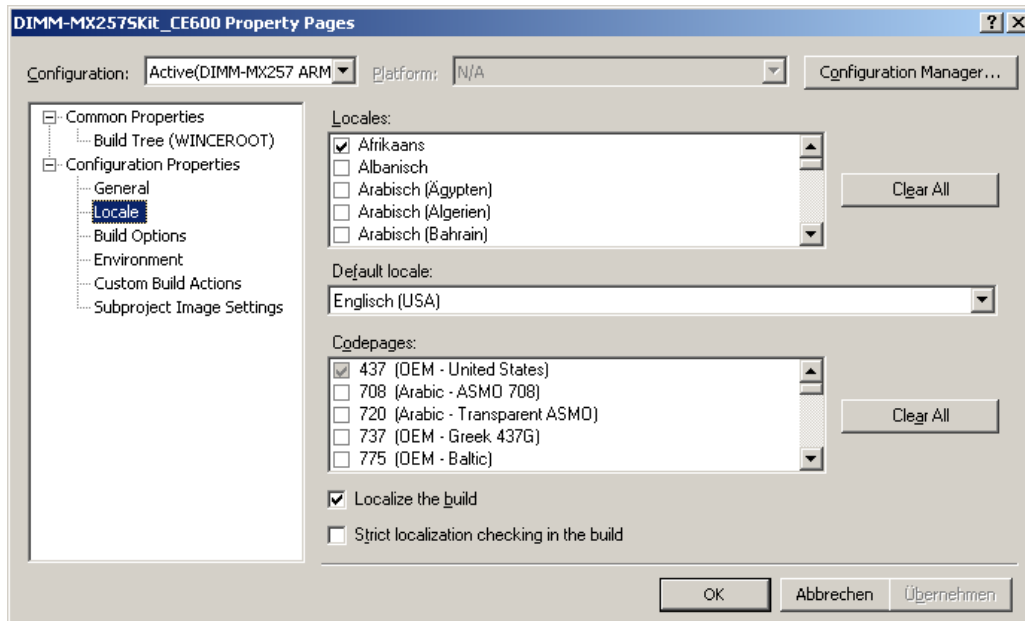


11.1 Settings of the Developer Kit kernel

Following the next sections you will get an overview about the settings of the kernel. In general, all settings have to be made before building the kernel.

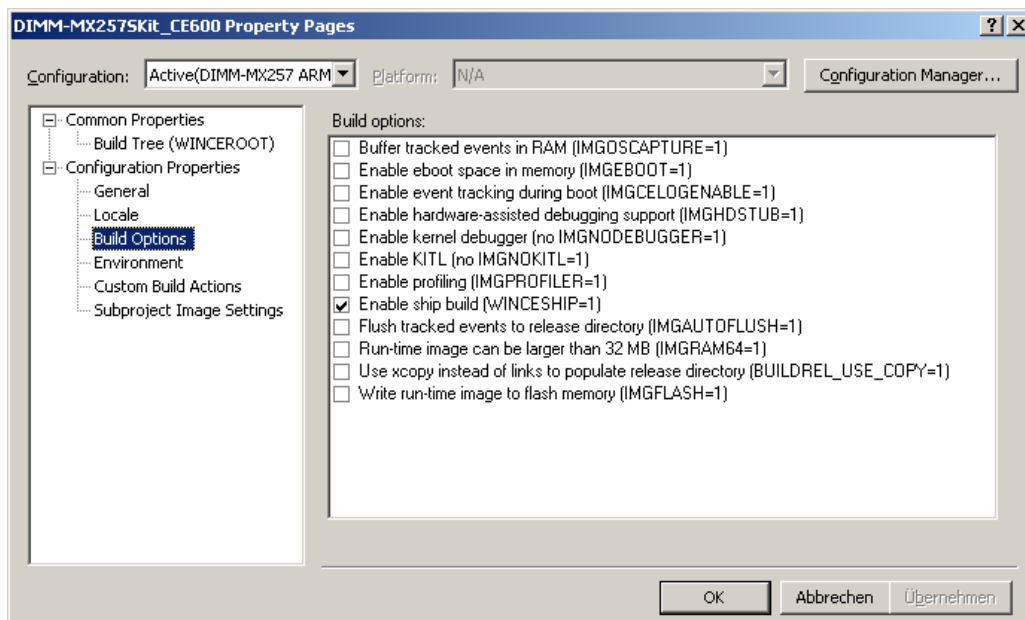
11.1.1 Language settings

The language settings of the kernel can be found in Locale.



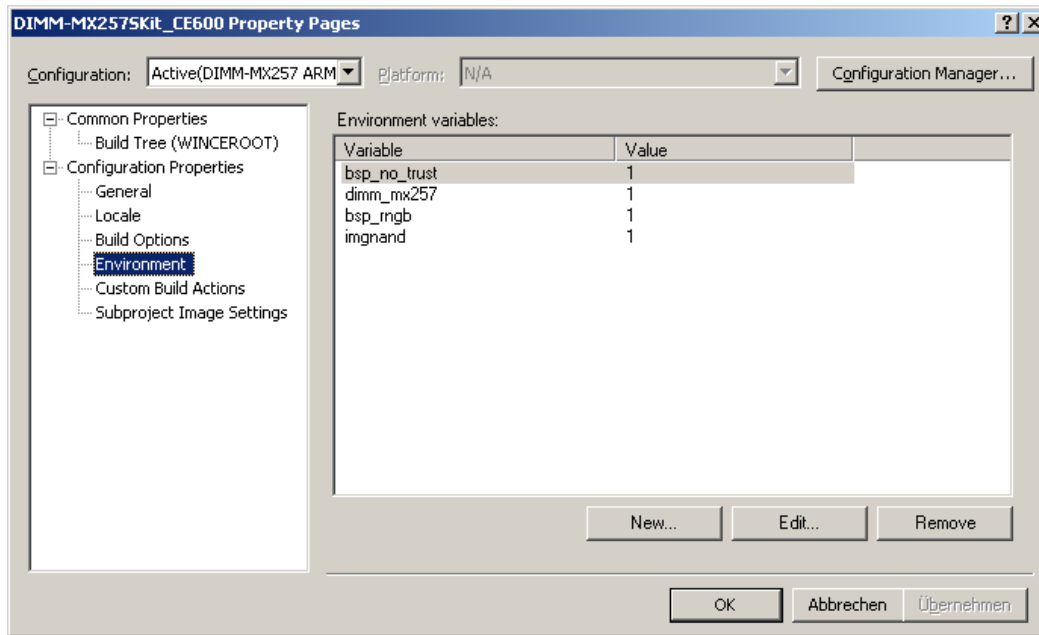
11.1.2 Build Options

There are many build options available. But for release, only one setting is enabled. This setting disables the output of messages from the kernel which use the RETAILMSG macro.



11.1.3 Environment settings

Following environment variables have to be set.

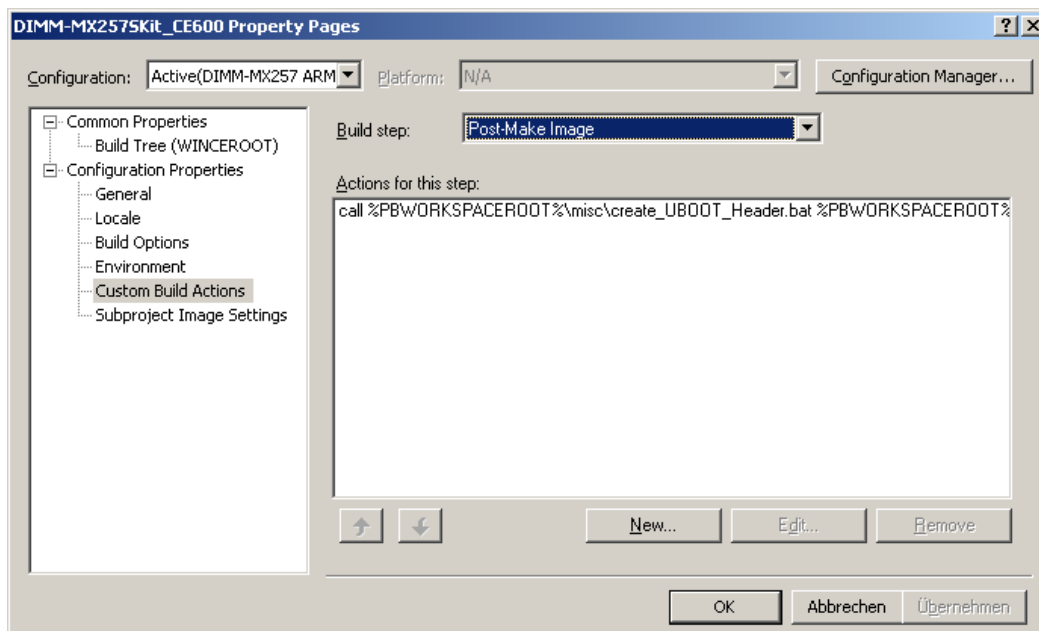


11.1.4 Custom Build Actions

The build process is using some custom batches for generating the kernel image of the Developer Kit. The batches do some special work in several of the build steps.

In the build steps "Pre-Sysgen" and "Pre-Make image" some files are copied to different locations.

In the build step "Post-Make Image" the batch "create_UBOOT_Header.bat" is called. The batch is using some tools coming with the platform builder. The batch is preparing the original image nk.bin to a special one for the bootloader. First, the raw binary nk.nb0 is created and then extended with a special header. Additionally, the image is licensed if a specified file containing the product key is present.



Remark :

If you create your own OSDesign you have to add the execution of the batch file "create_UBOOT_Header.bat" to the "Post-Make Image" step. Otherwise, you can not download the created image to the target device.

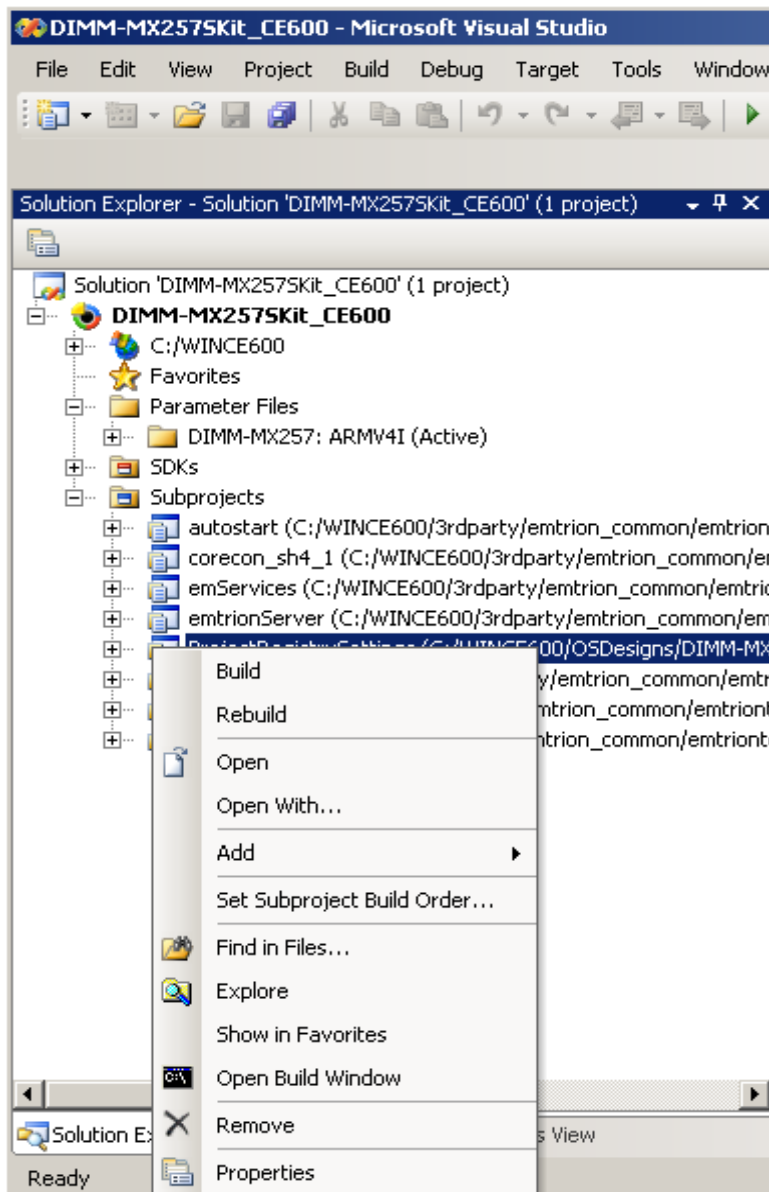
11.2 Build

Whenever you are building a kernel of the Developer Kit, the build order of the subproject described in the following section must be considered.

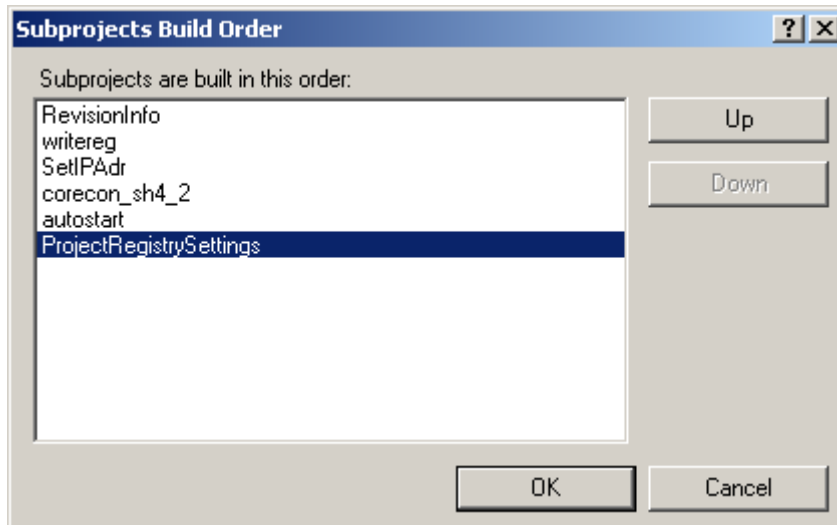
11.2.1 The subproject ProjectRegistrySettings

The OSDesign of the kernel includes some subprojects and also the subproject

ProjectRegistrySettings. This project contains some important registry entries and is provided for adding, removing or overwriting of any existing entry. These registry entries have to merged in a defined build order. This project must be build as last at any time.



The subproject ProjectRegistrySettings is built at last

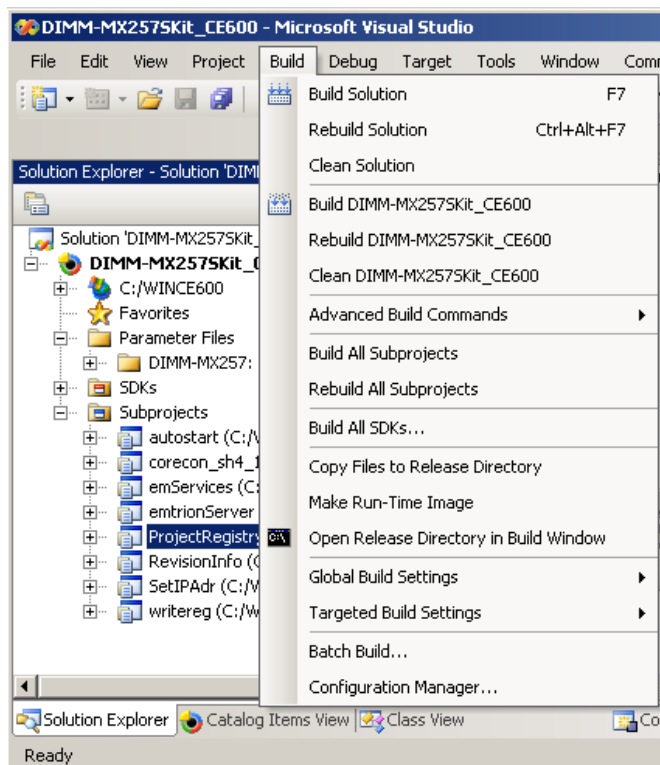


11.2.2 Starting the build process

A complete build process is required in the following cases:

- At first build of the kernel
- Adding or removing drivers
- Adding or removing OS components
- Changing (Switching) the OSDesign
- After you have update the BSP from the emtrion update repository

In all cases choose [Build] -> [Clean Solution] and the [Build] -> [Build Solution] from the menu.



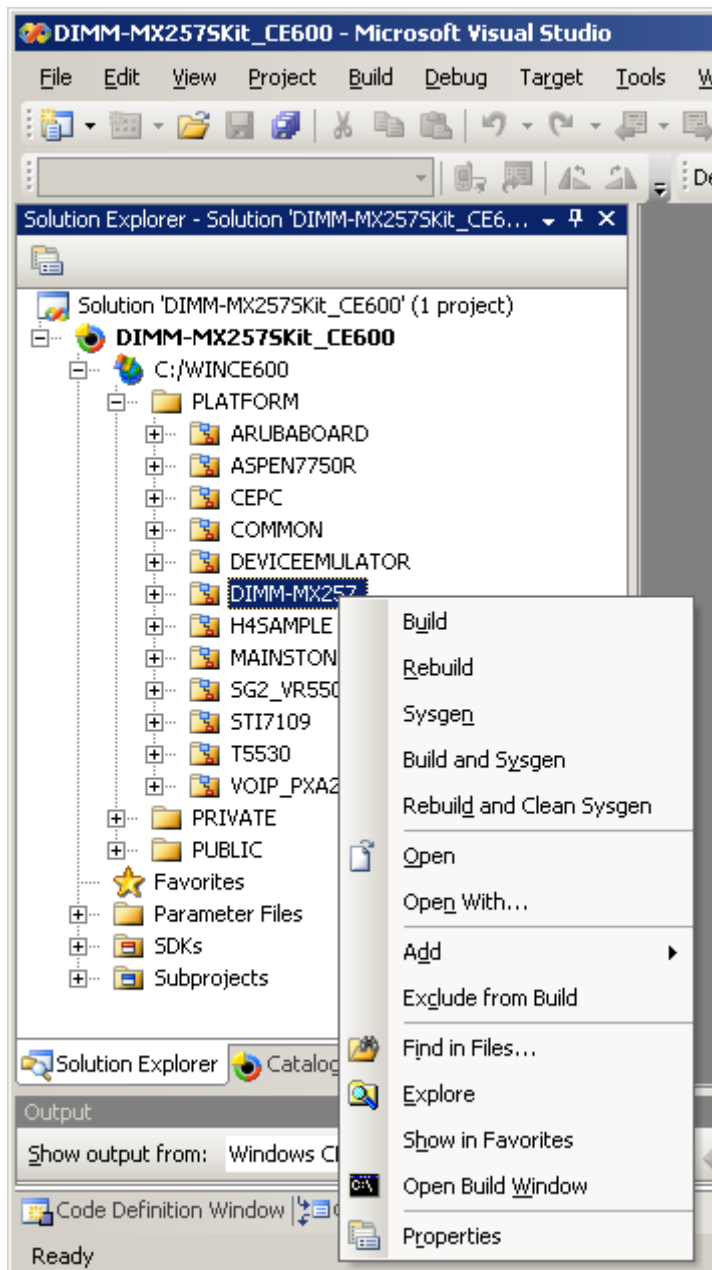
11.2.3 Incremental build

If only source code and build settings are modified an incremental build is sufficient.

At this point the checkboxes of [Global Build Setting] and [Targeted Build Settings] from the “Build” menu must be checked.

From the “Solution Explorer” select the appropriate component and right click on the mouse to open the context menu. Select build or rebuild to start the build process.

The figure below shows the incremental rebuild of the Developer Kit’s Platform, for example due to changing of the build settings.



12 Licensing of the WINCE-Image

When a licensed WINCE-Image should be created, it is necessary to use the mechanism created by emtrion. This mechanism stamps the WINCE-Image at the end of the build process automatically before the required header for the bootloader is added. For that, a simple text file must exist. This text file contains the product key which you got from Microsoft's license label (you can use the same key for all items of the same device type if you stick a license label on each device).

If the text file with the product key found the batch file "create_UBOOT_header.bat" creates both types of images: a licensed and an unlicensed one. The output file names are wce- dimm-mx257 for the licensed one and wce-dimm-mx257-eval for the unlicensed one.

The name of the text file is defined to "image_pid_ce600" and its location is the same as for "create_UBOOT_Header.bat". Please look for the location in the figure in section "**Fehler! Verweisquelle konnte nicht gefunden werden.**"

Example of the input format for the Product Key in "image_pid_ce600"

AAACD-FG8KJ-KL9NO-P2RST-UV5X3

Remark:

The key shown above is not a not valid key. You must use the key which you get from one runtime license label for Windows Embedded CE 6.

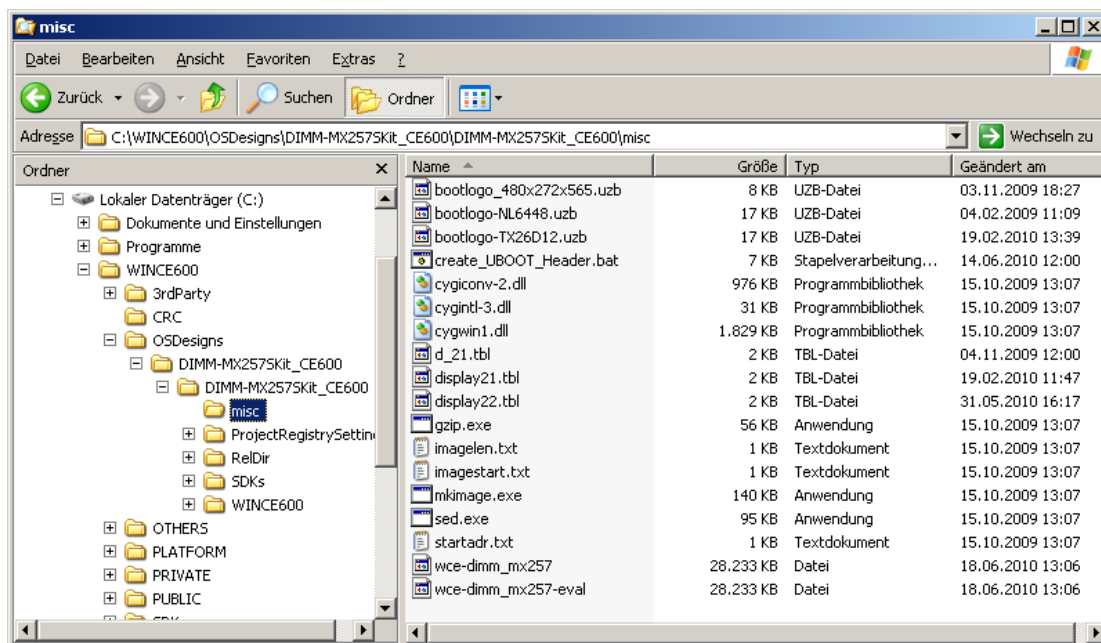
13 Downloading the image to the target

The build process in the previous sections generates the image of the OSDesign. But this image cannot be downloaded to the target because the bootloader requires a special header to identify an image.

To satisfy the bootloader, the batch “create_UBOOT_Header.bat” is running at the end of the build process. See also in [Custom Build Actions](#). It creates an extended image based on the origin one and adds a special header. Furthermore the image is renamed according to the conditions in the section [Licensing of the WINCE-Image](#).

In the case if only an unlicensed WINCE-Image is created, the output name has manually to be renamed to the definition of the environment variable “wimg” of the bootloader. This is normally the name of the licensed WINCE-Image.

Location of the extended image and batch



13.1 Perform download

Downloading to the target is explained in the Bootloader Manual.

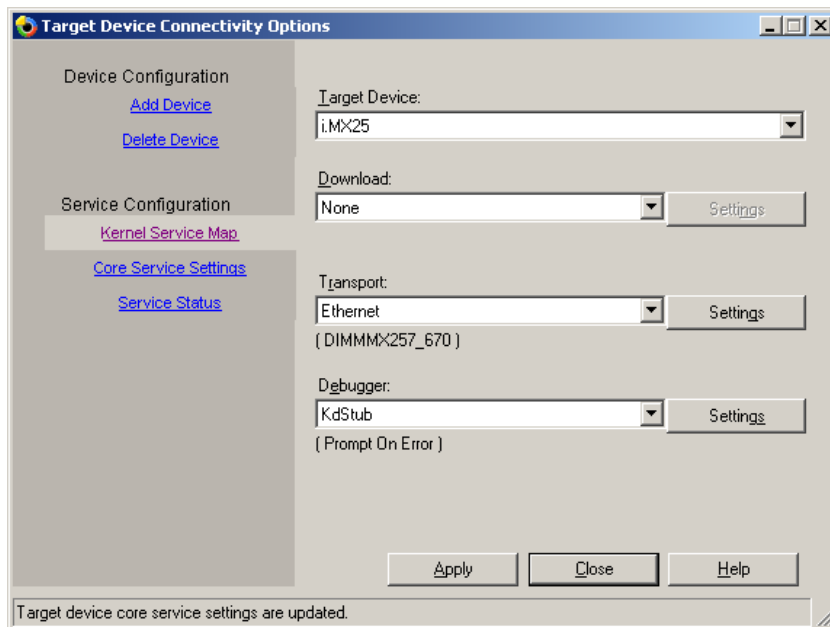
14 Setting up the Platform Builder for target connection

A debug as well a release version can be debugged by the Platform Builder. For them, the image has to be built with the build options “Kernel Debugger” and “KITL” enabled.

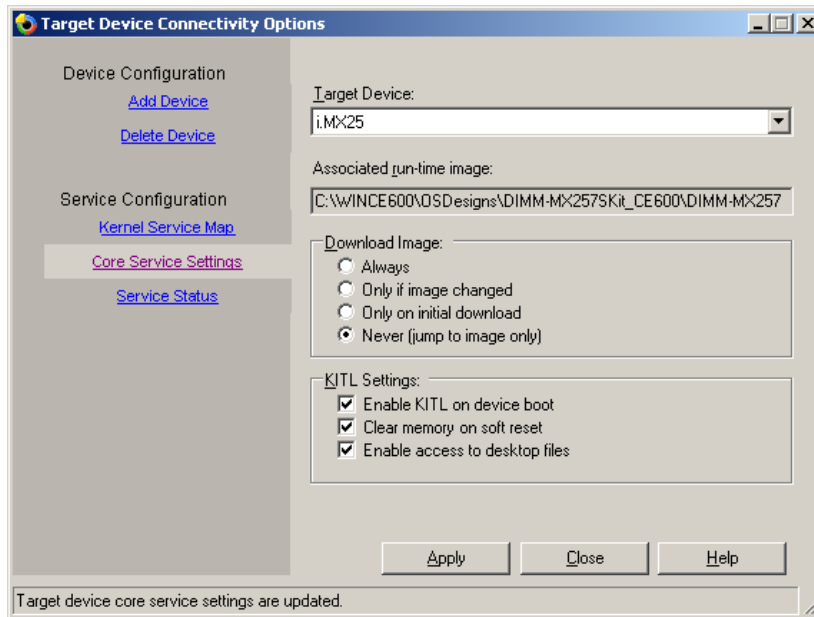
This section informs how you can establish a connection between Platform Builder and target for debugging purpose.

From the steps following below, the steps 4 to 10 have only to be done once.

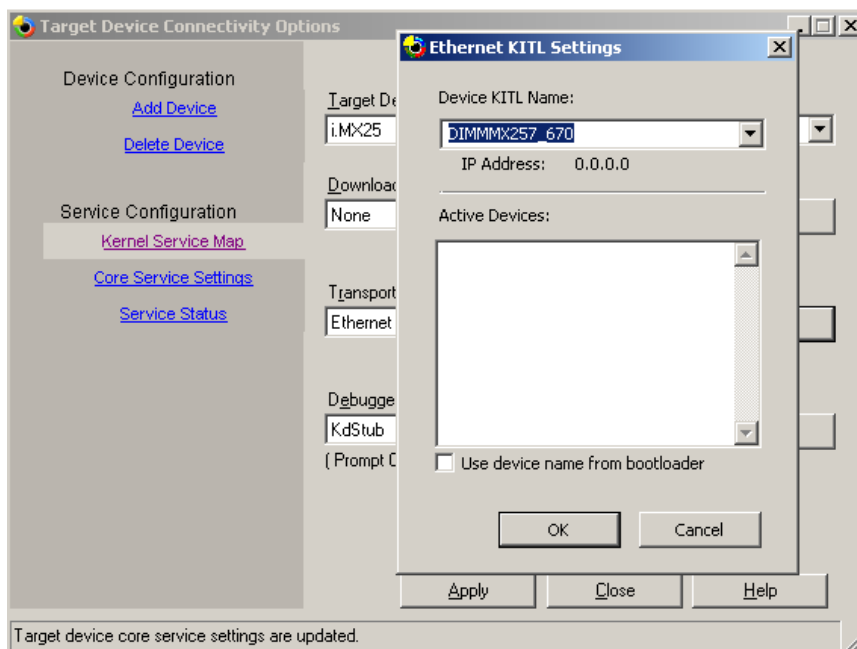
- (1) Make sure the OSDesign of the kernel is open.
- (2) Build the image with Kernel Debugger and KITL enabled.
- (3) Download the image to the target and store it in RAM
- (4) From the “Target” menu of VS2005 select “Connectivity Options ...” to open the “Target Device Connectivity Options” dialog box.



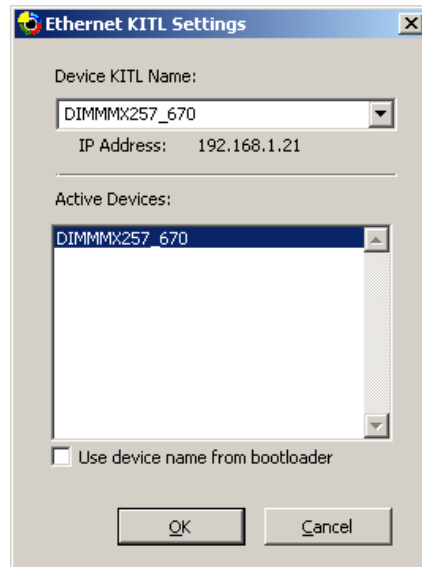
- (5) Under “Service Configuration”, choose “Core Service Settings”. Make the settings for “Download Image” and “KITL Settings” as shown below and apply the changes.



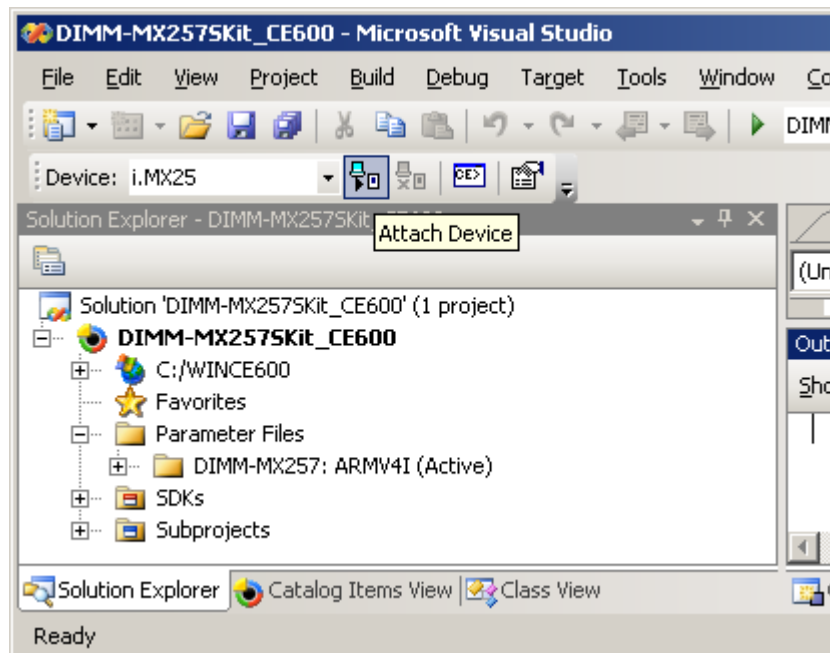
- (6) Go to "Kernel Service Map". Make the settings for the list boxes as in the dialog box in (4) above.
- (7) Choose the associated "Settings" button for "Transport". The Platform builder is waiting for the device. From the "Ethernet KITL Settings" dialog the device name of DIMM-MX257 is displayed as "DIMMMX257_" followed by numerals, which are the result of decimal conversion of the lower-order two byte of the LAN controller's MAC address.



- (8) When the KITL on the target has connected to the Debugger of the Platform Builder the list "Active Devices" is updated by the device name of your target DIMMMX257_.... See the figure below. Then select the device name in the list "Active Devices". The IP address of the target will appear below the list "Device KITL_NAME:".



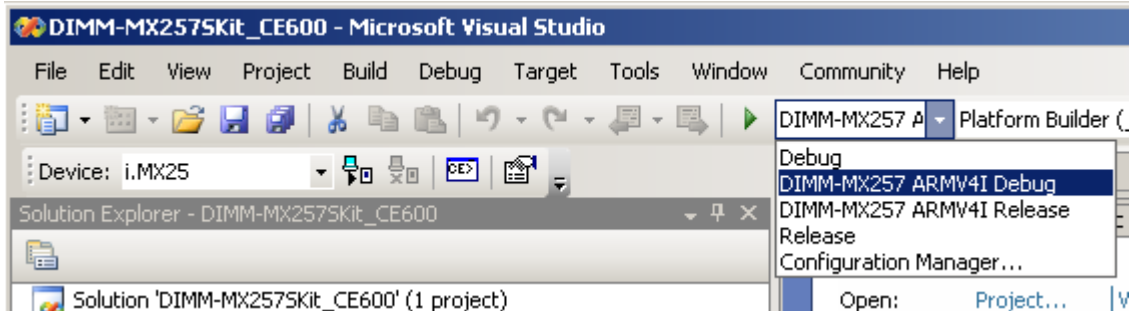
- (9) Press "OK" button to leave the "Ethernet KITL Settings" dialog box.
- (10) Apply the "Target Device Connectivity Options" dialog box and close it.
- (11) From the "Target" menu or "tools bar" select "Attach Device". The Platform Builder will connect with the Target.



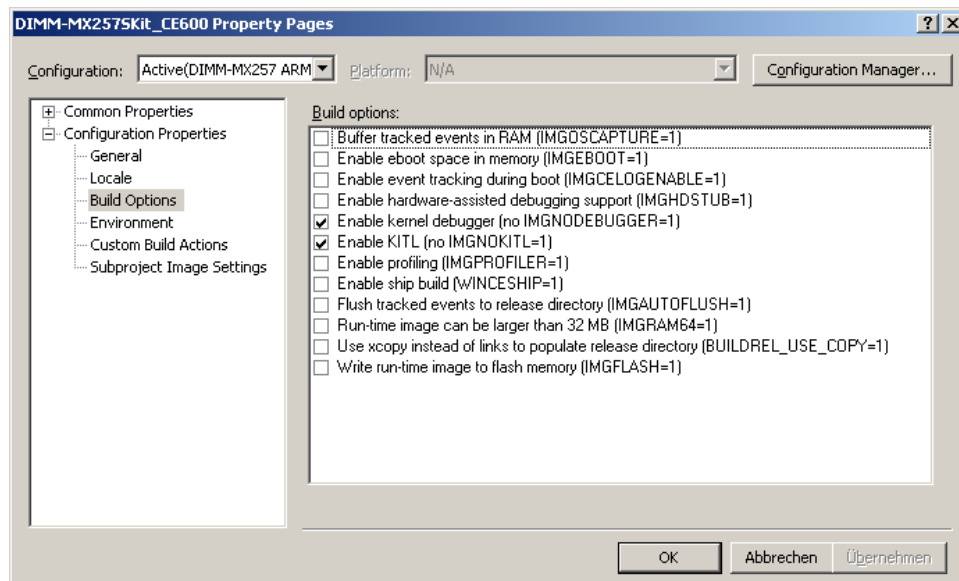
15 Building a Debug WinCE-Image (Kernel Development)

This section informs about the option settings made for the Debug version of the OSDesign of the Developer Kit kernel.

- (1) Make sure that the OSDesign of the Developer Kit kernel is open
- (2) From the "Solution Configurations" field, select "DIMM-MX257 ARMV4I Debug".



- (3) Open the Project Properties pages
- (4) The language sections are the same as for release Version in [Language settings](#).
- (5) The build options differ to release version. Kernel Debugger and KITL are enabled.



- (6) The "Environment" and "Custom Build Actions" are the same as for release Version.
- (7) Leave the Project Properties pages and build the image by following the steps in section [Build](#).

16 Data Exchange between Target and Workstation

The Developer Kit kernel allows the exchange of data between the target and your workstation. For this, Microsoft's ActiveSync application is needed. The latest version is available on the internet at <http://search.microsoft.com/search/results.aspx?st=b&qu=ActiveSync&view=en-us>

If you have not worked yet with Microsoft's ActiveSync, we recommend to take a look at the section "16.4 Remarks on Using ActiveSync".

16.1 Preparing the Desktop Computer

If Microsoft ActiveSync has not yet been installed on your desktop computer, it must be prepared as follows:

- 1.) Start the setup program and follow the shown instructions.

16.2 Preparing the Windows CE Device

USB function is the default setting for communication with ActiveSync. This communication is sufficient for file transfer and for application debugging.

When you do application development by Visual Studio 2005 communication via Ethernet is also possible. In this case USB ActiveSync can be used to get the IP address of the device.

16.3 USB ActiveSync

- Using a USB cable to connect the device to the workstation
- When not yet powered on, turn on the power on the target. The Windows® Embedded CE OS kernel which is stored in the flash will be executed. The running kernel will initiate an ActiveSync connection at the desktop computer. When the target is unknown as USB-Device at the workstation, Windows® will ask for the corresponding USB-function-driver which has to be installed. The driver is located in the subdirectory USBFDriver on the DVD.
- When ActiveSync does not start automatically on the workstation then you must perform this manually. Please note, USB has to be selected in the connectivity settings of ActiveSync.

The target establishes a connection with the desktop computer and possibly prompts you to login and enter your password. Enter the same name and password as you are using for your desktop computer.

Note:

When the connection is **not** automatically established between the target and desktop, you should start the "repllog" application on the target manually. To do this, select "run" from the "Start" menu, then enter **repllog** and click the **OK** button.


The "New Partnership" dialog appears on the desktop computer.

- Select the options as described in section "16.4 Remarks on Using ActiveSync"

- After establishing a connection, click Explore to open a window. Via this window, you may then exchange files with the target using drag and drop. For this, drag the files onto the desired directory on the target.

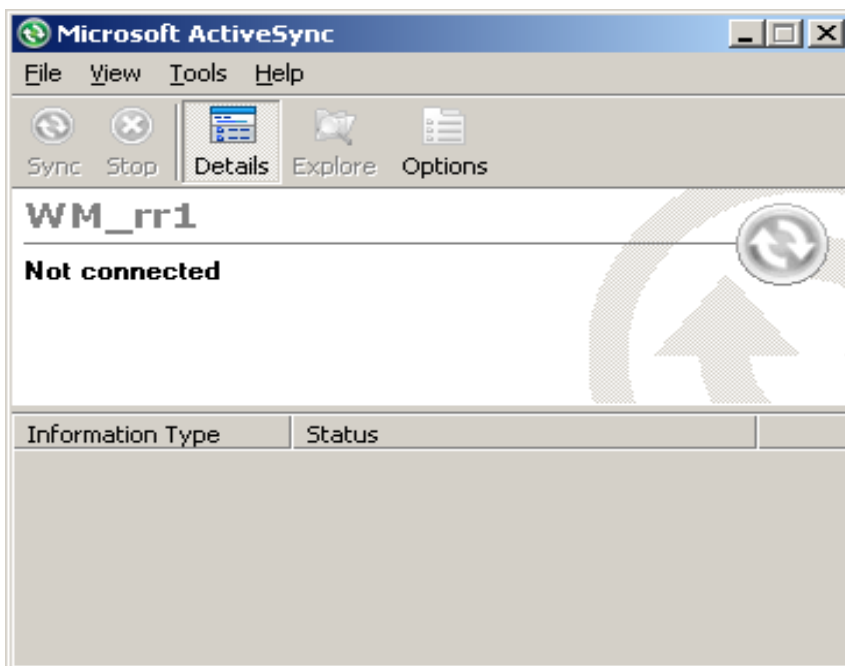
Note:

The folders of the kernel and their contents are created when Windows® Embedded CE is starting. IF you copy any files to these folders, they are not available any more when Windows® Embedded CE is restarted again. This does not apply to folders and files which are directly located at mass storages like "the "\FlashDisk" or the "\Storage Card". These folders are available only if the corresponding devices are detected in the system during the start up phase. The subfolders of the "Network" folder are a special case. These folders represent drives when a connection via network was established.

To terminate the connection, click the  symbol on the Windows® Embedded CE task bar. The connection can be detached in the dialog that appears. Another easy way to stop ActiveSync is just to unplug the USB cable.

16.4 Remarks on Using ActiveSync

After installing ActiveSync, there is a "Microsoft ActiveSync" link in the "Programs" group of the Windows "Start" menu. When pointing to this link, the following window appears:



Please verify that the USB connection is enabled ("Connection Settings" menu item of the "File" menu). If a connection has been established for the first time after power on, the following dialog box is shown:



The window shows that no partnership has been established between the Windows® Embedded CE computer and the desktop PC.

Hint:

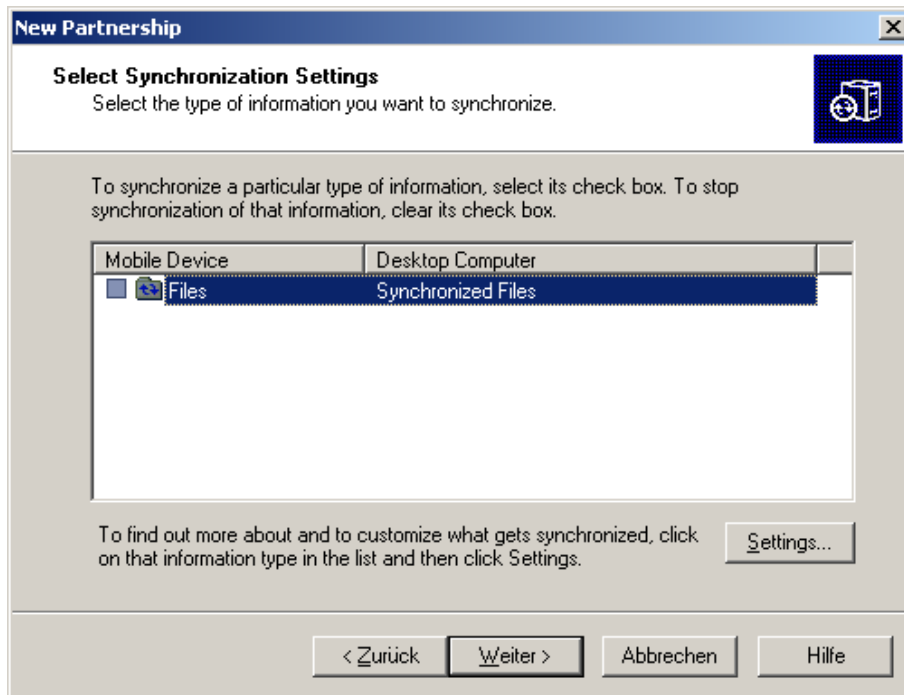
If a partnership has already been created with a device with persistent registry, the dialogs displayed here do not show up.

Any of the offered options will be using the USB interface with following property.

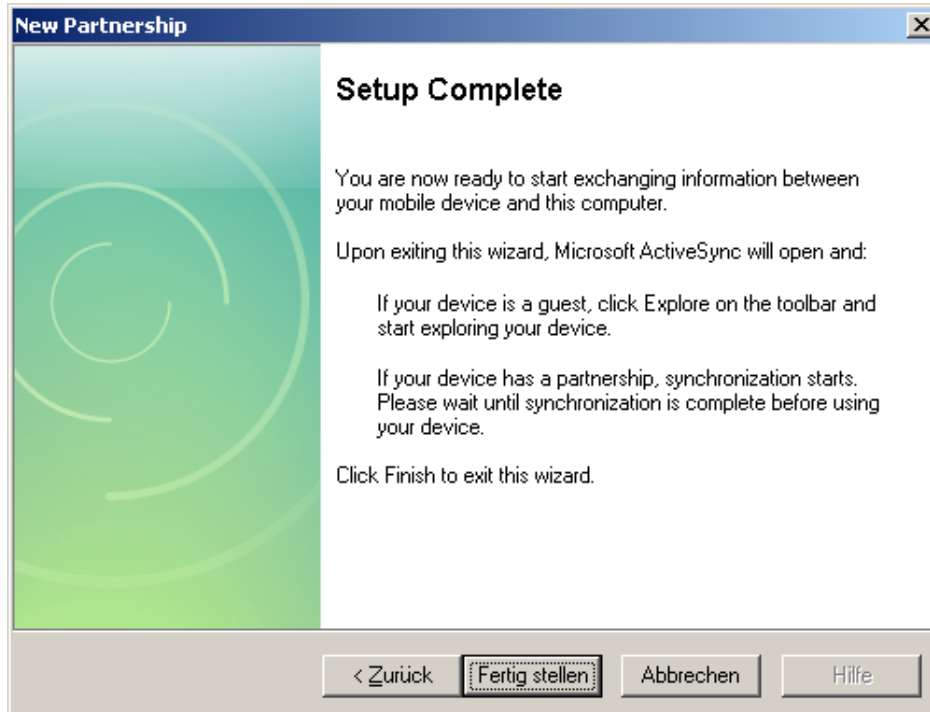
“No” means that the Windows® Embedded CE computer is connected with the desktop computer as a guest only. The connection as a guest is sufficient if you wish to exchange data or to debug an application via an USB connection.

“Yes” will establish a partnership, which is necessary for data synchronization.

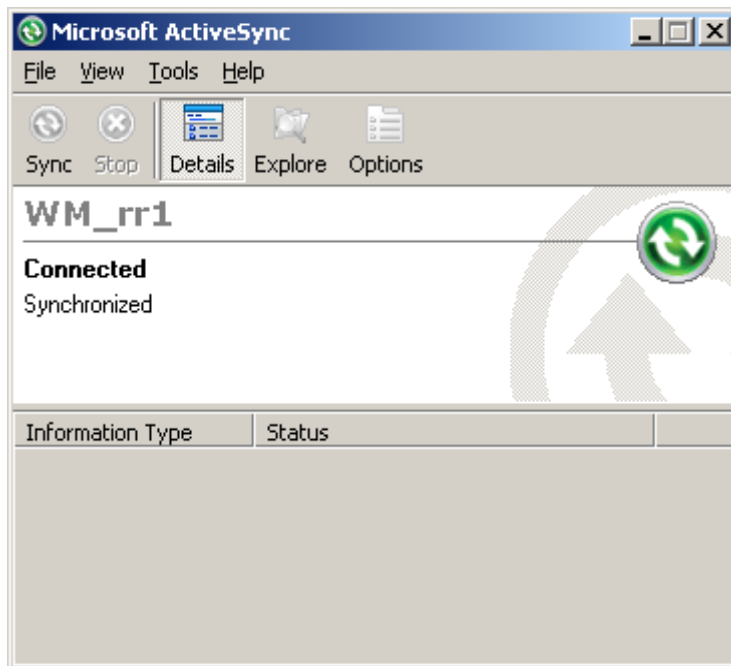
The following dialog appears:



Accept the default setting or deactivate "Files" by clicking on the small box left of "Files" and click "Next".



After that, a dialog box appears telling you that the partnership has been successfully established. After clicking the "Finish" button, the ActiveSync application will open on your desktop:



The connection has been established.

16.4.1 Data transfer

Clicking the "Explore" button will bring up a window where you can search the Windows® Embedded CE computer.

The data exchange between the Windows® Embedded CE computer and the desktop computer can be initiated by dragging a file from the window to another window that does not belong to ActiveSync, or vice versa.

17 Application development

Application development is only possible with VS2005 and VS2008. Development with Embedded Visual C++ is not supported with Windows Embedded CE 6.0 kernel. If there is a need for this, please don't hesitate to contact emtrion GmbH.

VS2005/VS2008 supports unmanaged and managed code and programming languages like C++, C#, Visual Basic and more are available. The communication interface for debugging can be Ethernet as well as USB ActiveSync. Sole Ethernet communication without a little help from USB ActiveSync is possible, but needs some special tools instead. For a more detailed description see the next chapters. We recommend a combination of Ethernet and USBF and all the kernels are supporting this.

17.1 VS2005 for Application Development

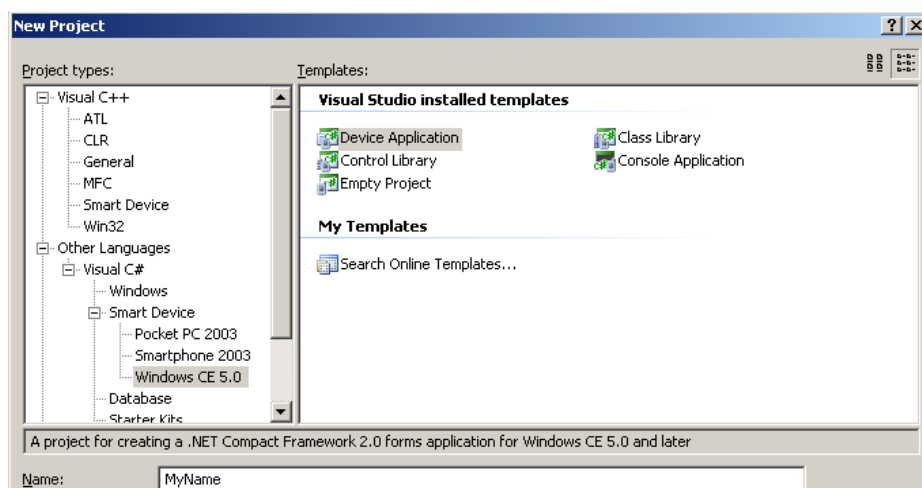
When you using VS2005 for application development based on a specific kernel some hints shall be considered.

- Since Win32-API functions may be omitted in the creation of operating system kernels under Windows® Embedded CE, there is no SDK (Software Development Kit) which suits all Windows® CE operating system kernels. Actually, each operating system kernel has an SDK of its own which must be installed in addition to the development environment.
- Please consult the online help of VS2005 (search for "Requirements") to obtain information on whether a specific function is available in an SDK. If the name of the OS kernel (e.g. DIMM-MX257DevKit_CE600_SDK for starter kit kernel) is listed, the function will be supported by this OS kernel.
- The list of parameters of the Win32-API functions is identical with that of the corresponding API functions of Windows® 9x/NT/XP. With some functions certain parameters must be set to 0. For more information, please refer to the online help.

17.1.1 Creating a new Managed Project

The first step is starting a new instance of VS2005.

1. Select **File/NewProject ...** from the Visual Studio menu.
2. In the **NewProject** window select **Visual C#/SmartDevice/WindowsCE5.0**
3. Select the **Device Application** template
4. Name your project **MyName** and click **ok**.



Now you can start with your application development.

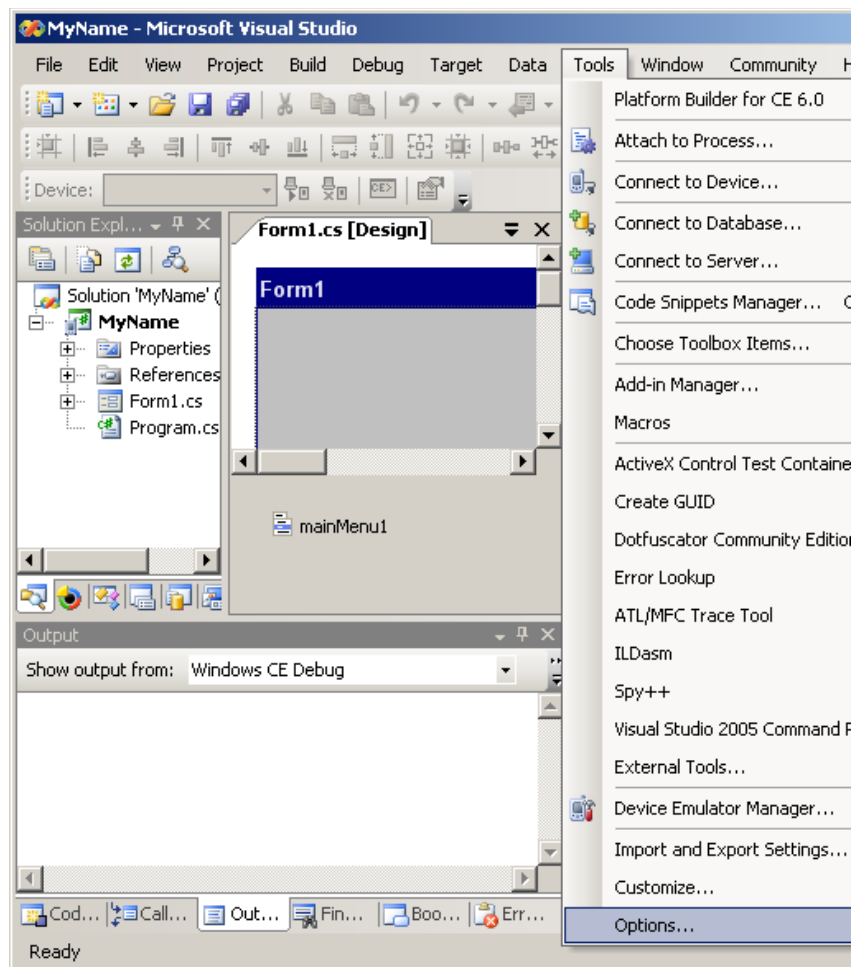
17.1.2 Deploying to the target

The connection between VS2005 and the target is made by network. For this reason the IP address of the target is necessary. To make the IP address known to VS2005 there are two options.

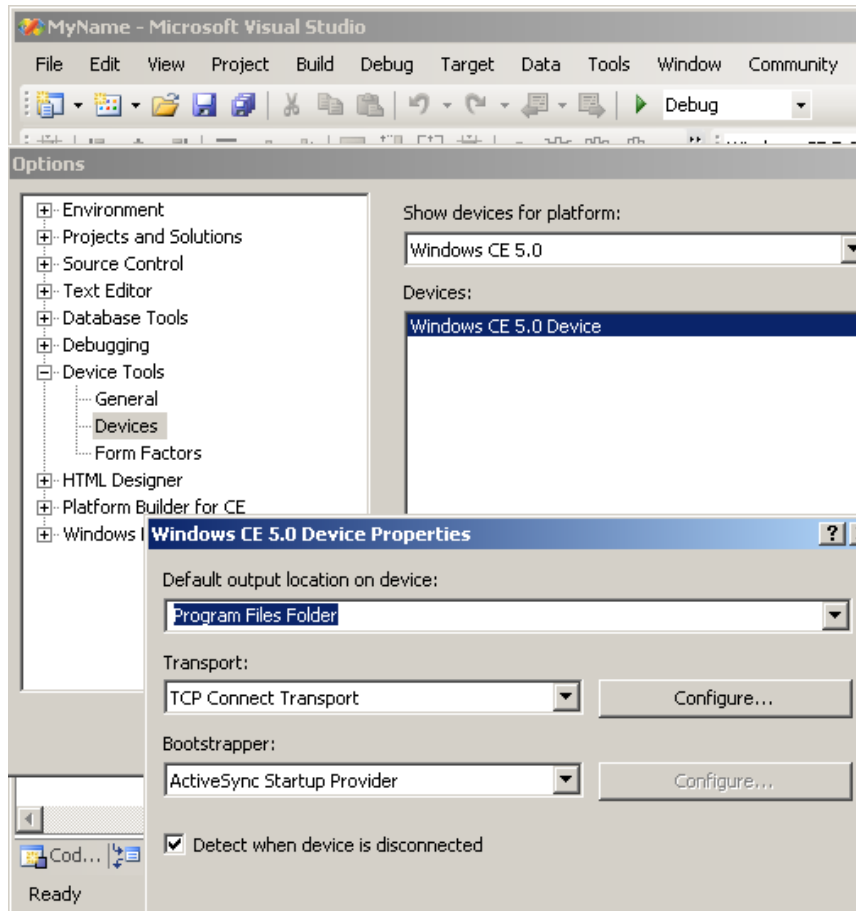
The first option is to ask manually by means of the function „ipconfig“ on the device. For that “ipconfig” must be executed from the command shell.

The second is receiving the IP address automatically by an existing USB ActiveSync connection. After getting this address, the next step is preparing the managed application development environment for deployment.

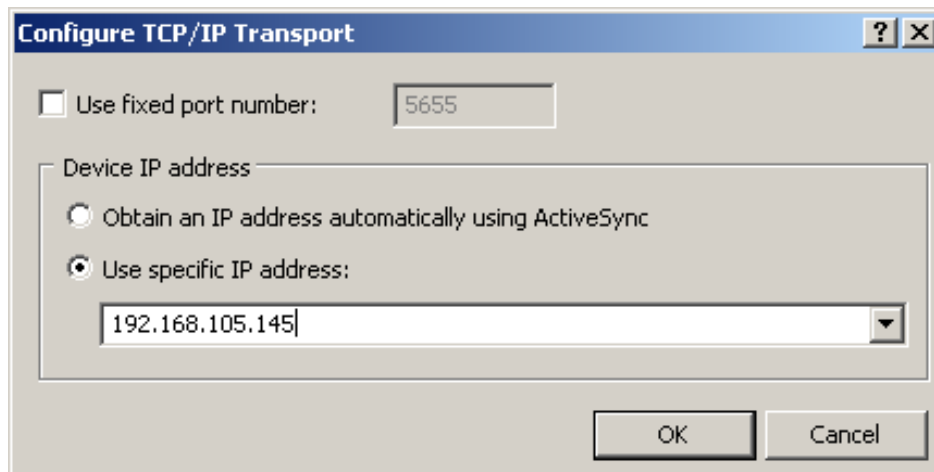
1. In VS2005 select **Tools/Options** from the menu.



2. In the options window, expand the **Device Tools** node and select **Devices**.
3. In the **Show device for platform:** drop down box select **WindowsCE5.0 Device**.
4. Click on **WindowsCE5.0 Device** and select **Properties**.



5. Click the **Configure** button beside the **Transport** drop down box. We are going to configure the TCP Connect Transport.



6. In the case when an USB ActiveSync connection exists between the workstation and the device, the option "Obtain an IP address ..." is to select. In the other case the option "Use specific IP address" is to select and the IP address you got on the device by ipconfig has to be inserted.
7. Click **OK** through all of the dialogs.

Preparing the target:

1. At the command shell, type **ConmanClient2**.
2. Then, type **cmaccept**. You have 3 minutes to establish a connection with your managed application.

Deploying the managed application:

1. Select **Debug/Start Debugging** from VS2005.
2. Select **WindowsCE5.0 Device** from the list of devices in the **Deploy MyName** box and click **Deploy**. VS2005 will deploy several cab files to the device in addition to your application. Your application is executed on the target device.

17.2 VS2008 for Application Development

In general, VS2008 is used in a very similar way as VS2005 for application development. So, for creating a project for a Windows CE Application and connecting to the target, please refer to the previous section.

If you want to install the SDK and have not also installed the Visual Studio 2005 on the same workstation as the Visual Studio 2008, you **must** choose the custom installation and deselect the option "Documentation". Otherwise you will get an installation error like this:

```
ToolsMsmCA(Error): IHxFilters filter registration failure Err=0x80040305, Context =  
pFilters->SetNamespace(Namespace)
```

For a coexisting installation of VS2005 and VS2008 you have to regard the following points:

- A coexisting installation is possible when Visual Studio 2008 is installed *after* Visual Studio 2005 without any known issues. If you reverse this installation order you get trouble when you developing applications for smart devices.
- Application debugging with Visual Studio 2005 when Visual Studio 2008 is also installed on the development machine:
In this case you have to consider the same points as you are using Visual Studio 2008 because with the installation of Visual Studio 2008 the host-side files are exchanged with a newer version (the files are in the same location on the development machine).
- How can I identify the version of the files used for the connection?
You can identify the file version on the development machine when you are using the Windows Explorer. Open this application, browse to the location of the files (\Program Files\Common Files\Microsoft Shared\CoreCon\1.0\Target\wce400\<CPU>) and select one of the executable files. Then press the right mouse button and select in the context menu the item "Properties". There is a tab named "Version". On this tab you see the file version. Starts the file version with 8.0.* the files comes from the Visual Studio 2005 installation. When the file version starts with 9.0.* than the files comes from the Visual Studio 2008 installation.

18 Starting an application at system start

There are two options to start an application at start up. Using only one option is allowed at any time. In any case, the execution of the explorer is controlled by emtrion's autostart mechanism described in the section below. This means no launch entry may be existed in the registry for the explorer.

To use the autostart functionality the registry must be modified. For it you can use the "Remote Registry Editor" of Visual Studio. For making the changes persistent, the emtrion tool "WriteReg.exe" is available.

Additional items have to be considered when you want to realize starting your application at system start:

- Folder structures and shortcuts get lost at power off or reset when it has been created at runtime.
- The same as above happens with files that have been copied to the object store of the kernel.
- Changes to the registry also getting lost when it have not been made persistent.
- Due to supporting autostart of applications from a storage device, the kernel is configured with search paths of any supported storage device.
- Dependent on the supported storage devices of the Developer Kit the storage folders are specified as following:
 - \SD Card for storage device microSD or SD card
 - \MultiMediaCard for storage device MMC
 - \Hard Disk for storage device USB stick
 - \FlashDisk for storage device NOR flash
 - \NAND flash for storage device NAND

18.1 Emtrion's autostart mechanism

One possibility to start you application at system start is using the autostart mechanism from emtrion. Therefore an autostart application was written and included into the kernel.

The registry of the Developer Kit kernel includes a launch link to that autostart application and is starting it at system start. Additionally, a shortcut of the autostart application is located in the "StartUp" folder of the explorer, so the startup mechanism of the explorer is also useable. Now, the autostart application itself looks in a specific registry key whether or not what application shall be started. The registry key includes several entries.

But is not the only task of the autostart application is also controlling the execution of the explorer.

The autostart application looks in

```
[HKEY_LOCAL_MACHINE\Software\emtrion\AutoStart]
```

for the entry "CustomerApp". The entry has the type REG_SZ. The entry provides the filename and if necessary the path of the application that shall be started. When the entry is empty no further action is taken. In the case of a valid entry the autostart application tries to start the specified application. When a path is not stated, the system searches in the folder "Windows" and in the

folders of the supported storage devices of the Developer Kit, mentioned in the introduction of this chapter.

The entry "AppParameter" can be found in the same registry key. The entry has also the type REG_SZ. This entry specifies the command line parameters of the application.

Optionally, the same registry key may also contain an entry called "WaitCycles". This entry, a type of REG_DWORD, lets you extend the time that the autostart application is waiting for the directories to be linked, e.g. a FlashDisk. If the entry is not found a default value of 100 is taken.

Optionally, the same registry key may also contain an entry called "FailureOptions". This entry has also the type REG_DWORD and is bit-coded. Please set only the bits that are really necessary. The meaning of the individual bits are shown in the following table:

Bit No.	Description
0	Setting this bit, a message is displaying in a message box, when the custom application cannot be started.
1	With this bit you can specify the message showing in the message box. 0: "Cannot start the application which was specified to the launch at startup" 1: more detailed information is showing, like the name and search path of the application.
2	Setting this bit, the autostart application doesn't exit and is showing continuously the message when the custom application cannot be started
3	Setting this bit, the explorer is started when the custom application cannot be started.
4	Specifies the start up process for the custom application 0 -> the explorer is not started and the custom application is started at launch time of the autostart application 1 -> the explorer is started and the custom application is started by the startup mechanism of the explorer
5	Setting this bit, the explorer will be started in any case
6	Setting this bit when the kernel is a headless one. Due to no graphic exists, the console is started instead.
7 . . 31	Reserved for future extensions.

A further entry is "Timeout". On that you can specify the time between two consecutive WaitCycles. The type of Timeout is REG_DWORD.

18.2 Autostart with LaunchXX and DependXX

This section describes how you can modify the kernel in order to start your application at system start by the two registry entries LaunchXX and DependXX.

Using this option, be sure the registry key "CustomerApp" in "18.1 Emtrion's autostart mechanism" is empty. In the case of using the explorer by the kernel, the registry key "FailureOptions" has to be set to 00000020h for execution.

The application and its own DLLs that shall be started have to be stored in one of the supported storage devices.

Here are the settings to do on the registry key

```
[HKEY_LOCAL_MACHINE\init]
```

- Entry 1:
 - 1.) Name: LaunchXX, where XX stands for a two-digit number. This number must be between 51 and 99 with the operating system kernel belonging to the Developer Kit kernel
 - 2.) Type: REG_SZ
 - 3.) Value: Name of the file that shall be executed.
- Entry 2:
 - 1.) Name: DependXX, where XX must be the same two-digit number as specified with entry 1.
 - 2.) Type: REG_BINARY
 - 3.) Value: Hexadecimal digits that specify the dependency on a previously started application.

Please also not the chapter "The Persistent Registry".

Note:

Entry 2 specifies the dependency of the application. An application stored on a mass storage cannot be started until the corresponding entry has been made in the folder structure. It may happen in this context that the registry entries in the [HKEY_LOCAL_MACHINE\init] key will be processed faster than the mass storages are entered in the folder structure. This must be taken into account in the start sequence.

19 Sample Applications

19.1 SKitApp

The sample application SKitApp is a simple terminal application that transmits all user inputs via the selected serial port. At the same time, receiving data is possible and is displayed in the main window of the application.

The communication settings (baud rate, number of data- and stop bits, parity, handshake) are displayed in the so-called "Command bar" (=combination of menu bar and tool bar) (see Figure). These parameters can be set via the menu:



The sample application was developed with Visual Studio 2005 and can be found on the Developer Kit DVD in the folder \Samples\SampleApp\SKitApp.

19.1.1 How the sample application work

After the application has been started, the window will be created and the global variables initialized. The application shows the serial ports (COM1... COMx) but only the available serial ports are highlighted. The application is now ready for user inputs.

Select the desired port from the "Parameters" -> "Port" menu.

As soon as a port is selected, it will be opened by calling the "OpenPort" function. The parameters chosen and the timeout time will be set. When the port is opened, a receive thread is created that receives the characters from the selected port and stores them in the receive buffer. Whenever a character has been received, a message with the WM_USER ID is transmitted to the main thread (primary thread). These WM_USER messages have the application repaint your window and so display the characters received.

A character is directly transmitted from within the main thread if it receives a message of the type WM_CHAR. This message is always received by the operating system when any key is pressed. The message handler calls the WritePort function that converts the character to be transmitted into ASCII code and then transmit it.

19.1.2 Changing the parameters via the menu

You are able to alter the serial port and the communication parameters in the "Parameters" menu. Each time a parameters is changed, the main thread gets a message of the type WM_COMMAND. The sub ID transmitted to the wParam parameter causes the corresponding parameters in the DCB to be modified. After this, the "UpdateTheSettings" function is called. This function sets the corresponding check marks with the menu items and updates the parameter string ("User Settings") displayed in the menu bar. "UpdateThe Settings" also calls the "UpdatePortSettings" function, in order to return the parameters of the port in use.

19.1.3 Changing the port

The port can be changed any time via the "Parameters" menu. If another interface is selected, the open port is closed (by calling "ClosePort"). After this, the global variable is corrected with the port

number, and the new port will be opened. When the new port is opened, it will be initialized with the same parameters as the last port has used.

20 The Persistent Registry

By default Microsoft's Windows® Embedded CE does not support a persistent registry. That is, all modifications made to the registry while Windows® Embedded CE is running will not be kept when Windows® Embedded CE is restarted.

To permanently keep these modifications, two functions are available which have to be implemented by the hardware manufacturer. These functions are already implemented in the kernel. They will be called by the operating system, in order to write the registry in hive-based form onto the Flash File System .

Note:

Systems with a non-persistent registry have the advantage that a system cannot be damaged due to possibly wrong registry settings made while the system is running.

In order to make the modifications persistent, the "RegFlushKey" function must be called by the application. However, this function should not be called after each modification to avoid performance bottlenecks. Call this function after having made several changes to the registry. The system also automatically writes changes back to the registry, but no exact statement can be made about this time interval. Therefore, use the "RegFlushKey" to ensure that the changes are saved.

The Developer Kit's operating system kernel includes a tool called WriteReg.exe This tool calls the RegFlushKey function to make the desired changes. For example, it can be used for system modifications that are to be kept permanently.

The command line

```
writereg -default
```

sets the registry to its default values, this means that all changes are lost and the settings of the default registry are used. A programming sample:

```
HKEY          hRegKey;
DWORD         retWert;

/* open the registry key */
retWert = RegOpenKeyEx(HKEY_LOCAL_MACHINE, _T("SOFTWARE"), 0, 0, &hRegKey);
if (retWert != ERROR_SUCCESS)
{
    /* Error handling */
}

/*
:
:
Here new entries are made, entries changed or deleted.
:
:
*/
```

```
/* Save registry to Flash */
retWert = RegFlushKey( hRegKey );
/* Hint: here a valid handle for the registry key will need to be specified
(see the online help). But always the complete registry will be saved to
flash */
If (retWert != ERROR_SUCCESS)
{
    /* Error handling */
}

/* Close registry key */
retWert = RegCloseKey( hRegKey );
if (retWert != ERROR_SUCCESS)
{
    /* Error handling */
}
```

20.1 Deleting the persistent registry

The persistent registry can be deleted in either of the following ways:

- 1.) Using the bootloader
- 2.) Via an application under Windows CE

20.1.1 Deleting the registry by the bootloader

The persistent registry can be deleted via bootloader as described in section [Reset Windows CE Registry](#).

20.1.2 Deleting the registry by an application under Windows Embedded CE

A Windows CE application can be enabled to use the persistent registry by calling the Win32-API function `KernelIoControl`. For this purpose, emtrion has created the following device-specific I/O control code:

```
#define IOCTL_HAL_SETREGISTRYTODEFAULT
        CTL_CODE(FILE_DEVICE_HAL, 0x810, METHOD_BUFFERED, FILE_ANY_ACCESS)
```

In this code, `CTL_CODE` is a macro, `FILE_DEVICE_HAL`, `METHOD_BUFFERED` and `FILE_ANY_ACCESS` are specified as `#define` statements. Both the macro and the definitions were defined by Microsoft in the `winiocctl.h` header file. This file is usually linked to applications with the

```
#include <windows.h>
```

statement.

20.1.2.1 IOCTL_HAL_SETREGISTRYTODEFAULT

This control code will delete the currently stored persistent registry. As a result, the default registry will be used when the system is started again. However, a reset will **not** be issued automatically. Such a reset can be performed by using the `IOCTL_HAL_REBOOT` control code.

IOCTL_HAL_SETREGISTRYTODEFAULT expects the following parameters of the KernelIoControl function.

Parameter	Description
lpInBuf, nInBufSize, lpOutBuf, nOutBufSize	These parameters will not be analyzed.
lpBytesReturned	Pointer to a UINT32(DWORD) variable containing the number of bytes returned (here 0)

The control code deletes the flash sectors (where the registry is saved) by means of the erase function supplied by the bootloader.

20.1.2.2 Documentation of the KernelIoControl function (extraction from the online help of Platform Builder)

This function provides the kernel with a generic I/O control for carrying out I/O operations:

Function prototype:

```

BOOL KernelIoControl( DWORD dwIoControlCode, LPVOID lpInBuf,
                     DWORD nInBufSize, LPVOID lpOutBuf,
                     DWORD nOutBufSize, LPDWORD lpBytesReturned );

```

Parameter	Description
dwIoControlCode	I/O control code, which should supported the OAL I/O controls.
lpInBuf	Pointer to the input buffer
nInBufSize	Size, in bytes, of lpInBuf
lpOutBuf	Pointer to the output buffer
nOutBufSize	Maximum number of bytes that can be returned in lpOutBuf
lpBytesReturned	Address of a DWORD that receives the size, in bytes, of the data returned.

Return value:

Return value	Description
TRUE	indicates success
FALSE	indicates failure

Remarks:

The kernel calls the OEMIoControl function when a device driver or application calls the kernel function KernelIoControl and passes an I/O control code. The system is fully preemptible when this function is called. The kernel does no processing, but it passes all parameters directly to the function supplied by you. This function is provided solely to allow your device driver or application to communicate with an OAL and its specific functionality.

21 Emtrion Tools for Windows Embedded CE

In order to do various settings or get information, emtrion provides a variety of tools running on the Windows Embedded CE computer. Some of the tools can be found in "Start" menu -> "Programs" -> "emtrion Tools".

21.1 Writing the persistent registry

In order to enable the user to selectively write to the registry, emtrion has implemented a tool called WriteReg in the operating system kernel. When starting, it writes the persistent registry by calling the Win32-API function RegFlushKey().

When it is called with the option "-default" the persistent registry is deleted and the OS is starting with its internal registry at next start.

Note:

User inputs will be of no significance while the persistent registry is being written.

21.2 Getting the Version of the Operating System Kernel

The RevisionInfo tool implanted in the operating system kernel helps the user to determine the currently running operating system kernel.

The following information is displayed in a dialog box:

- Name of the operating system kernel (= name of the OSDesign), e.g. DIMM-MX53_CE700
- Version number of the operating system kernel, e.g. V1
- Release date of the operating system kernel, e.g. 2011/26/02

21.3 SetIPAdr

This tool is not displayed within the folder structure stated above. The tool setipadr is a console application and is equipped with some command line options. You can get current information about the network adapter or you can set some properties like the IP address or the subnet mask.

Calling the tool without any options shows a list of all the available network adapters:

Option	Value	Description
-a	Name of the network adapter	Displays the current network settings of the specified adapter
-d	Specify 0 or 1	Disables or enables the DHCP for the specified adapter in -a
-h	No value is required	Available parameters are listed
-i	Specify the IP address in dot notation	Sets the IP address for the specified adapter in -a
-n	Specify the subnet mask in dot notation	Sets the network mask for the specified adapter in -a

Example:

This command line disables the DHCP and set the IP address and network mask for the network adapter TEST:

```
setipadr -a TEST -d 0 -I 192.168.110.98 -n 255.255.255.0
```

22 Frequently Asked Questions (FAQ)

This section provides FAQs on the following

- In General

The FAQs presented here were up to date at the time this manual was being printed. Since they are continuously updated, you may wish to check our support site

<http://www.support.emtrion.de/> to get the latest FAQ on a variety of products. If you cannot find your product, or if the information provided is not sufficient, contact our support team (see also section "23 Support" for details).

22.1 In General

- How to determine the MAC address (=unique identification of each node in a network) of the target?

The MAC address of the target is hold in the environment variable "ethaddr" of the bootloader. From the bootloader prompt type printenv and then look for the specified variable. You will see the MAC address assigned to, like

```
ethaddr= 00:30:6C:90:00:10
```

- How to establish a network connection between a CE computer and a Windows® NT/2000/XP computer?

Method 1

1.) Open Internet Explorer on the Windows® Embedded CE computer.

2.) Example:

The "program files" directory is to be mapped. It is located on drive C: on a computer named NTRechner. The drive C: has the share name LW_C. In this case, enter "[\\NTRechner\LW_C\program](#) files" to the address bar.

3.) If the network is running, the contents of that drive appear.

- Method 2

1.) Open a command line prompt under Windows Embedded CE

2.) Enter the following command line:

```
net use <localname> <remotename> /user:<uname>  
/password:<pword>
```

The command line parameters have the following meaning:

Parameter	Description
<localname>	Name with which the mapped subdirectory appears in the “\Network” directory
<remotename>	Network path in accordance with the UNC naming convention for the drive to be mapped. Example: The “program files” directory is to be mapped. It is located on drive C: on a computer named NTRechner. The drive C: has the share name LW_C. In this case, enter “\\NTRechner\LW_C\program files” to the address bar
<uname>, <pwd>	User name (Login) and password on the computer connected to the network. Enter the same name and password as you are using for your desktop computer.

- 3.) In the “\Network” directory on the Windows® Embedded CE computer, there is now a new directory with the specified name. This directory represents the corresponding directory on the NT computer.

- The following message appears on the screen: “Auto Download Failed: Insufficient memory in object store to download to target device”

Dividing the RAM into a data- and program memory is not sufficient for your application. Open the System applet of the Control panel. Select the Main Memory tab. Move the slider to the right. This will increase the data memory. After this, you should be able to download your application.

- No DHCP available
By default, the operating system kernel of the Developer Kit are configured in such a way that the settings (IP address, subnet mask, gateway) are taken from a DHCP server.

Change these settings as follows:

- 1.) Open the “Network and Dial-up Connections” applet in the Control Panel.
- 2.) Select the corresponding network adapter
- 3.) Press the right mouse button and select “Properties” in the context menu.
- 4.) Select the “IP address” tab
- 5.) Select “Specify an IP address”
- 6.) Enter an IP address and the related subnet mask. Enter the IP address of your desktop computer as default gateway.
- 7.) Close the dialog box with OK.
- 8.) Save the persistent registry.
- 9.) Press the reset button on the target.
- 10.) Check the network connection as follows:
 - Open the Internet Explorer
 - Enter the following address
\\<name> of your desktop computer\<share name of one of your drives>

Example:

Name of the desktop computer: NTRechner

Share name of your C: drive: C\$

Address to be entered: [\\NTRechner\C\\$](#)

The name of the desktop computer can be obtained by entering "IPCONFIG /ALL" in the DOS prompt of the desktop computer

23 Support

This product has been thoroughly tested over the development period. Due to its complexity, however, no guarantee can be given that it will seamlessly operate under any circumstances. We are therefore grateful for any feedback regarding an incorrect operation of the boards.

If any problems should occur, have a look at the FAQ section of this manual first. Or visit our support web site <http://www.support.emtrion.de/> for the latest updates. Please check also if there already updates available in the update repository (see chapter "9.2 Updating the BSP to the newest release" for details).

To accelerate the process, please fill out the supplied form, which can be found on our web site at http://www.emtrion.com/support_form_en.php

To handle a support request we need at least the following information:

- Who sent the request (company name and the name of the writer)
- A valid email address, where we can send the answer
- The product name DIMM-MX53DevKit_CE700
- The product version and release date, which is printed on the CD (NOT the version of the Windows CE operating system).
- If you have updated the product from the update repository, the revision number of the version which you have downloaded
- A detailed description of the problem and how it can be reproduced
- Recommended: all applications which are required for reproducing your issue and which are not included in this Developer Kit.

24 License Agreement

24.1 Emtrion License Agreement

End User License Agreement

Emtrion GmbH Greschbachstrasse 12, 76229 Karlsruhe, Germany (hereinafter referred to as "**Emtrion**") provides to businesspersons which are "Unternehmer" in the sense of Sect. 14 BGB German Civil Code (hereinafter: "the **Customer**") certain Software to be run as embedded software on hardware boards (hereinafter: "the Software") subject to Emtrion's Terms and Conditions:

I. Definitions

1. "**Software**" shall mean the operating software to be embedded into the Hardware, including any documentation hereto.
2. "**Confidential Information**" shall mean any information, know-how and data owned or controlled by Emtrion or by Third Parties license Rights.

II. Limited Use Rights

With respect to Software, Customer is granted a non-exclusive, non-transferable, non-sub licensable, worldwide, license to use or have used the Software for validating or testing. Against an extra fee and an advanced EULA Customer can use Software for projects.

Customer shall not - and shall prevent others from doing so - copy, translate, modify, create derivative works, disassemble, reverse engineer, decompile or otherwise use the Software, except as specifically authorized hereunder in this item, or except provided otherwise by compulsory law. Whenever compulsory law permits the above actions, Recipient shall in any case

- (i) inform Emtrion of these actions in advance, and
- (ii) request support of Emtrion, which Emtrion may offer against reasonable consideration.

All rights granted to Software hereunder shall only apply to the code format to be provided as agreed upon, which is, if not explicitly agreed upon otherwise, solely the binary code to the Software.

Customer is aware that Software may consist of or contains third party software, distributed by Emtrion in altered or unaltered form. The use of such Software may be further restricted by additional terms and conditions (hereinafter also "**Third Parties License Rights**"), with priority to the terms set out herein.

III. **ADDITIONAL RESTRICTION ON USE**

THE OBJECT CODE, SOURCE CODE AND THIRD PARTY MATERIALS ARE NOT FAULT TOLERANT AND ARE NOT DESIGNED, MANUFACTURED, OR INTENDED FOR USE OR RESALE AS ONLINE CONTROL EQUIPMENT IN HAZARDOUS ENVIRONMENTS REQUIRING FAIL-SAFE PERFORMANCE, SUCH AS IN THE OPERATION OF NUCLEAR FACILITIES, AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH THE FAILURE OF THE OBJECT CODE, SOURCE CODE AND THIRD PARTY MATERIALS COULD LEAD DIRECTLY TO DEATH, PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE.

IV. **Confidentiality Restrictions**

1. **Restrictions.** Without limiting and in addition to the use restrictions under Sect. II and Sect III above, all Confidential Information delivered pursuant to this agreement shall
 - (a) be used by Customer for the Purpose and subject to Section II above only; and
 - (b) be treated by Customer with the same degree of care to avoid unauthorized disclosure to any third party as with respect to Customer's own confidential information of like importance but with no less than the degree of care described in this agreement; and
 - (c) not be distributed, disclosed, or disseminated in any way or form by Customer to anyone except its own employees, who have a reasonable need to know such Confidential Information with respect to the Purpose and who are bound to confidentiality by written agreements not less stringent than under the obligations of this agreement; and
 - (d) be stored in secure areas only that are protected against unauthorized access and are regularly monitored to assure their sufficient security,
 - (e) be treated by Customer in compliance with any additional security guidelines the parties may agree.
2. **Exceptions.** The obligations as per Sect. IV. 1 above shall not apply to any information which the Customer can prove,
 - (a) is at the time of disclosure already in the public domain or becomes available to the public through no breach by the Customer of this agreement, except that Confidential Information shall not be deemed to be in the public domain merely because any part of the Confidential Information is embodied in general disclosures or because

individual features, components or combinations thereof are now or become known to the public;

- (b) is received by the Customer from a third party free to lawfully disclose such information to Customer;
- (c) was in the Customer's lawful possession prior to receipt from the Discloser as evidenced by written documentation;
- (d) is independently developed by the Customer without the benefit of any of the Confidential Information as evidenced by written documentation;
- (e) is approved for release by written agreement of Emtrion;
- (f) is required to be disclosed to comply with legal mandatory regulations, a judicial or official order or decree, provided that written advance notice of such judicial action was timely given to Emtrion.

V. Prohibition on Development of Competing Products

Customer shall at no time during the term of this agreement and hereafter for a term of five (5) years use Confidential Information for the purpose of developing or distributing products which are competing with Emtrion's Hardware as a stand alone product.

VI WARRANTIES.

Emtrion makes no warranties, either express or implied, with respect to the object code or source code or with respect to Software or other products or intellectual property obtained from Third Parties. Emtrion expressly disclaims any such warranties, including but not limited to the implied warranties of merchantability and fitness for a particular purpose.

24.2 Microsoft License Agreement

SOFTWARE LICENSE TERMS

WINDOWS EMBEDDED CE 6.0 CUSTOMIZED OEM SOFTWARE DEVELOPMENT KIT for

Windows Embedded CE 6.0 Developer Kit for HiCO.DIMM7723, 02.0.0

These license terms are an agreement between you and [emtrion GmbH] "Company". Please read them. They apply to the software named above which includes software licensed by Microsoft Corporation or their affiliates ("Microsoft") to Company. The software also includes documentation and any media on which you received it.

The terms also apply to any

- updates,
- supplements,
- Internet-based services, and
- support services

for this software, unless other terms accompany those items. If so, those terms apply.

By using the software, you accept these terms. If you do not accept them, do not use the software. Instead, contact Company [www.emtrion.com] to determine its return policy. If you comply with these license terms, you have the rights below.

- 1. INSTALLATION AND USE RIGHTS.** You may install and use one copy of the software on each computer on your premises to design, develop, test and demonstrate your programs running validly licensed copies of Microsoft Visual Studio 2005, Team Suite, Visual Studio 2005 Team Edition for Software Architects, Visual Studio 2005 Team Edition for Software Developers, Visual Studio 2005 Team Edition for Software Testers, Visual Studio 2005 Professional Edition, Visual Studio 2005 Tools for the Microsoft Office System or Visual Studio 2005 Standard Edition (including evaluation versions).
- 2. SPEECH RECOGNITION.** If the software includes speech recognition component(s), you understand that speech recognition is an inherently statistical process and that recognition errors are inherent in the process. Neither Company nor Microsoft or their suppliers shall be liable for any damages arising out of errors in the speech recognition process.
- 3. SCOPE OF LICENSE.** The software is licensed, not sold. This agreement only gives you some rights to use the software. Company and Microsoft reserve all other rights. Unless applicable law gives you more rights despite this limitation, you may use the software only as expressly permitted in this agreement. In doing so, you must comply with any technical limitations in the software that only allow you to use it only in certain ways. For more information, see the software documentation. Except and only to the extent permitted by applicable law despite these limitations, you may not:
 - work around any technical limitations in the software;
 - reverse engineer, decompile or disassemble the software;
 - make more copies of the software than specified in this agreement;
 - publish the software for others to copy;
 - rent, lease or lend the software;
 - transfer the software or this agreement to any third party; or
 - use the software for commercial software hosting services.
- 4. BACK UP MEDIA.** You may make one backup copy of the software. You may use it only to reinstall the software.
- 5. TRADEMARKS.** This agreement does not grant you any rights in connection with any trademarks or service marks of Company, Microsoft, or their suppliers.
- 6. EXPORT RESTRICTIONS.** The software is subject to United States export laws and regulations. You must comply with all domestic and international export laws and regulations that apply to the software. These laws include restrictions on destinations, end users and end use. For additional information, see www.microsoft.com/exporting.
- 7. PRODUCT SUPPORT.** Contact Company for support options. Refer to the support number provided with the device. Microsoft does not provide support services for components provided to you in this software.
- 8. HIGH RISK ACTIVITIES.** The software is not fault-tolerant and is not designed, manufactured or intended for any use requiring fail-safe performance in which the failure of the software could lead to death, serious personal injury or severe physical and environmental damage ("High Risk Activities"), such as the operation of aircraft or nuclear facilities. You agree not to use, or license the use of, the software in connection with any High Risk Activities. Company customized this software and is responsible for how it operates.

9. Entire Agreement. This agreement, and the terms for supplements, updates, Internet-based services and support services that you use, are the entire agreement for the software and support services.

10. Applicable Law.

United States. If you acquired the software in the United States, Washington state law governs the interpretation of this agreement and applies to claims for breach of it, regardless of conflict of laws principles. The laws of the state where you live govern all other claims, including claims under state consumer protection laws, unfair competition laws, and in tort.

Outside the United States. If you acquired the software in any other country, the laws of that country apply.

11. Legal Effect. This agreement describes certain legal rights. You may have other rights under the laws of your country. You may also have rights with respect to the party from whom you acquired the software. This agreement does not change your rights under the laws of your country if the laws of your country do not permit it to do so.

12. NO WARRANTIES FOR THE SOFTWARE. The software is provided “as is”. You bear all risks of using it. Microsoft gives no express warranties, guarantees or conditions. When allowed by your local laws, Company and Microsoft exclude implied warranties of merchantability, fitness for a particular purpose and non-infringement. If your local laws give you any implied warranties, guarantees or conditions, despite this exclusion, your remedies are limited to Company and its suppliers using reasonable efforts to repair or replace the software to the extent permitted by your local laws. Any warranties you receive regarding the software do not originate from, and are not binding on, Microsoft or their affiliates.

13. LIMITATION ON AND EXCLUSION OF REMEDIES AND DAMAGES. You can recover from Microsoft and their suppliers only direct damages up to U.S. \$5.00. Except for any refund Company may provide, you cannot recover any other damages, including consequential, lost profits, special, indirect or incidental damages.

It also applies even if

- anything related to the software, services, content (including code) on third party internet sites, or third party programs, and
- claims for breach of contract, breach of warranty, guarantee or condition, strict liability, negligence, or other tort to the extent permitted by applicable law.
- Company or Microsoft knew or should have known about the possibility of the damages.

The above limitation may not apply to you because your country may not allow the exclusion or limitation of incidental, consequential or other damages.

Please note: As this software is distributed in Quebec, Canada, some of the clauses in this agreement are provided below in French.

Remarque : Ce logiciel étant distribué au Québec, Canada, certaines des clauses dans ce contrat sont fournies ci-dessous en français.

EXONÉRATION DE GARANTIE. Le logiciel visé par une licence est offert « tel quel ». Toute utilisation de ce logiciel est à votre seule risque et péril. L'entreprise ou Microsoft n'accorde aucune autre garantie expresse. Vous pouvez bénéficier de droits additionnels en vertu du droit local sur la protection des consommateurs, que ce contrat ne peut modifier. La ou elles sont permises par le droit locale, les garanties implicites de qualité marchande, d'adéquation à un usage particulier et d'absence de contrefaçon sont exclues.

LIMITATION DES DOMMAGES-INTÉRÊTS ET EXCLUSION DE RESPONSABILITÉ POUR LES DOMMAGES. Vous pouvez obtenir de Microsoft et de ses fournisseurs une indemnisation en cas de dommages directs uniquement à hauteur de 5,00 \$ US. Cette indemnisation ne couvre pas les réductions qu'une entreprise vous ait données. Vous ne pouvez prétendre à aucune indemnisation pour les autres dommages, y compris les dommages spéciaux, indirects ou accessoires et pertes de bénéfices.

Cette limitation s'applique aussi à :

- tout ce qui est relié au logiciel, aux services ou au contenu (y compris le code) figurant sur des sites Internet tiers ou dans des programmes tiers ; et
- les réclamations au titre de violation de contrat ou de garantie, ou au titre de responsabilité stricte, de négligence ou d'une autre faute dans la limite autorisée par la loi en vigueur.
- Ce que l'entreprise ou Microsoft savaient ou auraient du savoir sur les dommages-intérêts.

Elle s'applique également, même si Microsoft connaissait ou devrait connaître l'éventualité d'un tel dommage. Si votre pays n'autorise pas l'exclusion ou la limitation de responsabilité pour les dommages indirects, accessoires ou de quelque nature que ce soit, il se peut que la limitation ou l'exclusion ci-dessus ne s'appliquera pas à votre égard.

EFFET JURIDIQUE. Le présent contrat décrit certains droits juridiques. Vous pourriez avoir d'autres droits prévus par les lois de votre pays. Le présent contrat ne modifie pas les droits que vous confèrent les lois de votre pays si celles-ci ne le permettent pas.

15. ADDITIONAL TERMS (if applicable).

25 Literaturverzeichnis

1. **Freescale Semiconductor Inc.** *i.MX25 PDK 1.0 Windows Embedded CE 6.0 Reference Manual*. 6 2009.
2. **emtrion GmbH.** *DIMM-MX257 CAN-SW - Software documentation of CAN driver for Windows embedded CE 6.0 R3*. 2010. Rev.1.